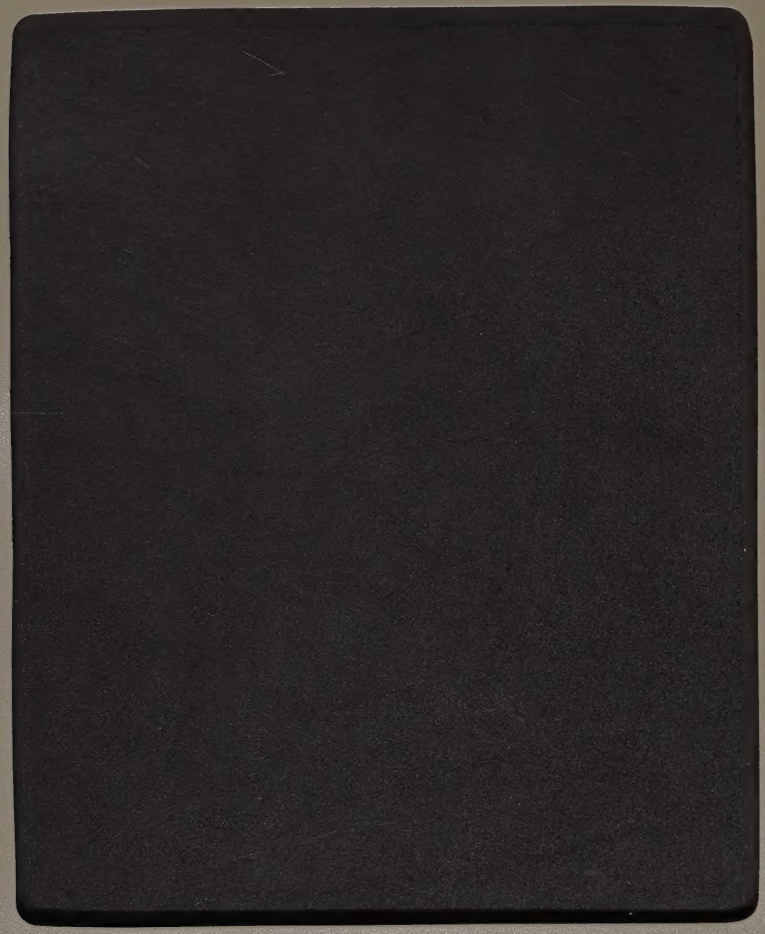
*03M01C15*

# 1720A
# Instrument Controller

-To RESTART
FROM Loop
Diag. Test

User Manual

FLUKE®



**List of Software Versions**

| | |
|---|---|
| BOOT | 1.2 |
| FDOS | 1.3 |
| CONMON | 1.1 |
| BASIC | 1.2 |
| FUP | 1.1 |
| SET | 2.0 |
| TIME | 1.0 |

NOTICE

This manual documents this set of software versions only. Software modules will function properly only in compatible versions. Refer to the software compatibility check procedure at the end of the installation section for additional information.

# 1720A Instrument Controller

CTRL-"C" - To Restart from Loop Diag. Test

## User Manual

P/N 518654
April 1980
Rev 1 11/80

©1980, John Fluke Mfg. Co., Inc., all rights reserved Litho in U.S.A.

FLUKE®

# WARRANTY

Notwithstanding any provision of any agreement the following warranty is exclusive:

The JOHN FLUKE MFG. CO., INC., warrants each instrument controller it manufactures to be free from defects in material and workmanship under normal use and service for the period of 90 days from date of purchase. This warranty extends only to the original purchaser. This warranty shall not apply to fuses, floppy disks or contents thereof, or any product or parts which have been subject to misuse, neglect, accident or abnormal conditions of operations.

In the event of failure of a product covered by this warranty, John Fluke Mfg. Co., Inc. will repair an instrument controller returned to an authorized Service Facility within 90 days of the original purchase; provided the warrantor's examination discloses to its satisfaction that the product was defective. The warrantor may, at its option, replace the product in lieu of repair. With regard to any instrument controller returned within 90 days of the original purchase, said repairs or replacement will be made without charge. If the failure has been caused by misuse, neglect, accident or abnormal conditions of operations, repairs will be billed at a nominal cost. In such case, an estimate will be submitted before work is started if requested.

THE FOREGOING WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS OR ADEQUACY FOR ANY PARTICULAR PURPOSE OR USE. JOHN FLUKE MFG. CO., INC., SHALL NOT BE LIABLE FOR ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, WHETHER IN CONTRACT, TORT OR OTHERWISE.

**If any failure occurs, the following steps should be taken:**

1. Notify the JOHN FLUKE MFG. CO., INC., or the nearest Service Facility, giving full details of the difficulty, and include the model number, type number, and the serial number. On receipt of this information, service data or shipping instructions will be forwarded to you.

2. On receipt of the shipping instructions, forward the instrument controller transportation prepaid. Repairs will be made at the Service Facility and the instrument controller returned, transportation prepaid.

**SHIPPING TO MANUFACTURER FOR REPAIR OR ADJUSTMENT**

All shipments of JOHN FLUKE MFG. CO., INC., instrument controllers should be made via United Parcel Service or "Best Way" prepaid. The instrument controller should be shipped in the original packing carton. If this original carton is not available a suitable container that is rigid and of adequate size may be used. However, the JOHN FLUKE MFG. CO., INC., does not recommend shipment of instrument controllers in substitute containers. In the event a substitute container must be used, the instrument controller should be wrapped in paper and surrounded with at least four inches of excelsior or other similar shock absorbing material. JOHN FLUKE MFG. CO., INC., shall assume NO risk for intransit shipment damage.

**CLAIM FOR DAMAGE IN SHIPMENT TO ORIGINAL PURCHASER**

The instrument controller should be thoroughly inspected immediately upon original delivery to purchaser. All material in the container should be checked against the enclosed packing list. The manufacturer will not be responsible for shortages against the packing sheet unless notified immediately. If the instrument controller is damaged in any way, a claim should be filed with the carrier immediately. (To obtain a quotation to repair shipment damage, contact the nearest Fluke Technical Center.) Final claim and negotiations with the carrier must be completed by the customer.

The JOHN FLUKE MFG. CO., INC. will be happy to answer all application or use questions, which will enhance your use of this instrument. Please address your requests or correspondence to: JOHN FLUKE MFG. CO., INC., P.O. BOX 43210, MOUNTLAKE TERRACE, WASHINGTON 98043, ATTEN: Sales Dept. For European Customers: Fluke (Nederland) B.V., Zevenheuvelenweg 53, Tilburg, The Netherlands.*

*For European customers, Air Freight prepaid.

**John Fluke Mfg. Co., Inc. • P.O. Box 43210 • Mountlake Terrace, Washington 98043**

Rev. 4/80/Controllers

# Warning

**This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with this User Manual, may cause interference to radio communications. As temporarily permitted by regulation it has not been tested for compliance with the limits for Class A computing devices pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.**

# Table of Contents


| SECTION | TITLE | PAGE |
|---|---|---|
| 1 | **INTRODUCTION** | **1-1** |
| | 1-1. DESCRIPTION OF 1720A DOCUMENTATION | 1-1 |
| | 1-3. INTRODUCTION TO THIS MANUAL | 1-3 |
| | 1-6. Command Definitions | 1-3 |
| | 1-8. Syntax Diagrams | 1-4 |
| | 1-11. THE 1720A INSTRUMENT CONTROLLER | 1-5 |
| 2 | **INSTALLATION AND START-UP** | **2-1** |
| | 2-1. INTRODUCTION | 2-1 |
| | 2-3. UNPACKING | 2-1 |
| | 2-6. Unpacking Checklist | 2-1 |
| | 2-7. Options and Accessories | 2-2 |
| | 2-9. INSTALLATION | 2-3 |
| | 2-11. Line Voltage Range Selection | 2-3 |
| | 2-16. Environment | 2-5 |
| | 2-21. CONTROLS, INDICATORS, AND CONNECTORS | 2-5 |
| | 2-23. START-UP PROCEDURE | 2-8 |
| | 2-25. SOFTWARE COMPATIBILITY CHECK | 2-12 |
| | 2-28. Compatible Versions of System Software | 2-12 |
| | 2-30. Procedure | 2-13 |
| 3. | **SYSTEM SOFTWARE** | **3-1** |
| | 3-1. INTRODUCTION | 3-1 |
| | 3-4. FDOS | 3-1 |
| | 3-6. CONMON | 3-1 |
| | 3-8. TIME | 3-1 |
| | 3-10. SET | 3-1 |
| | 3-12. FUP | 3-1 |
| 3A | **FDOS** | **3A-1** |
| | 3A-1. INTRODUCTION | 3A-1 |
| | 3A-3. DEVICE NAMES | 3A-1 |
| | 3A-5. THE SYSTEM DEVICE | 3A-1 |
| | 3A-8. FILE NAMES | 3A-2 |
| | 3A-12. COMMAND FILES | 3A-3 |
| | 3A-19. The System Command File | 3A-4 |
| | 3A-21. Utility Command Files | 3A-4 |
| | 3A-25. Command File Examples | 3A-6 |
| | 3A-34. Creating Command Files | 3A-7 |
| | 3A-38. CONTROL CODES | 3A-9 |
| 3B | **CONMON** | **3B-1** |
| | 3B-1. INTRODUCTION | 3B-1 |




*(continued on page ii)*

**CONTENTS** *continued*


| SECTION | | TITLE | PAGE |
|---|---|---|---|
| | 3B-3. | ACCESS TO CONMON | 3B-1 |
| | 3B-5. | Start-Up Access | 3B-1 |
| | 3B-8. | Access From Utility Programs | 3B-1 |
| | 3B-10. | Access From BASIC | 3B-2 |
| | 3B-12. | Direct Access | 3B-2 |
| | 3B-14. | USING CONMON | 3B-2 |
| **3C** | **TIME** | | **3C-1** |
| | 3C-1. | INTRODUCTION | 3C-1 |
| | 3C-3. | ACCESS TO TIME | 3C-1 |
| | 3C-5. | EXCITING TIME | 3C-1 |
| | 3C-7. | USING TIME | 3C-1 |
| **3D** | **SET** | | **3D-1** |
| | 3D-1. | INTRODUCTION | 3D-1 |
| | 3D-3. | ACCESS TO SET | 3D-1 |
| | 3D-5. | EXITING SET | 3D-1 |
| | 3D-7. | USING SET | 3D-1 |
| | 3D-11. | SET Command Structure | 3D-2 |
| | 3D-14. | Device Specification | 3D-3 |
| | 3D-17. | Setting Baud Rates | 3D-4 |
| | 3D-20. | Setting Character Length | 3D-5 |
| | 3D-23. | Defining a Parity Bit | 3D-5 |
| | 3D-28. | Defining Character Spacing | 3D-6 |
| | 3D-31. | Defining Terminator Characters | 3D-6 |
| | 3D-36. | Listing Port Configuration | 3D-7 |
| | 3D-40. | The SET Help Display | 3D-8 |
| | 3D-42. | SET ERROR MESSAGES | 3D-8 |
| | 3D-44. | SET COMMAND SUMMARY | 3D-8 |
| **3E** | **FUP** | | **3E-1** |
| | 3E-1. | INTRODUCTION | 3E-1 |
| | 3E-3. | ACCESS TO FUP | 3E-1 |
| | 3E-5. | EXITING FUP | 3E-1 |
| | 3E-7. | USING FUP | 3E-1 |
| | 3E-9. | FUP Command Structure | 3E-1 |
| | 3E-12. | ASCII Transfer (no option) | 3E-3 |
| | 3E-15. | Assign the System Device (/A) | 3E-5 |
| | 3E-18. | Binary Transfer (/B) | 3E-5 |
| | 3E-23. | Deleting Files (/D) | 3E-6 |
| | 3E-28. | Format, Verify, and Zero File Device(s) (/F) | 3E-6 |
| | 3E-32. | Listing a Directory (/L or /E) | 3E-7 |
| | 3E-36. | Merging ASCII Files (/M) | 3E-8 |
| | 3E-46. | Renaming a File (/R) | 3E-9 |
| | 3E-50. | Scanning For Bad Blocks (/S) | 3E-10 |
| | 3E-54. | Transferring ASCII Files Without Error Checks (/T) | 3E-10 |
| | 3E-59. | Whole Copying a File Device (/W) | 3E-11 |
| | 3E-66. | Zeroing a File Directory (/Z) | 3E-12 |
| | 3E-70. | The FUP Help File | 3E-13 |
| | 3E-72. | FUP ERROR MESSAGES | 3E-13 |
| | 3E-74. | FUP COMMAND SUMMARY | 3E-14 |
| **4** | **HARDWARE** | | **4-1** |
| | 4-1. | INTRODUCTION | 4-1 |
| | 4-3. | SYSTEM ARCHITECTURE | 4-1 |


*(continued on page iii)*

**CONTENTS** *continued*


| SECTION | TITLE | PAGE |
|---|---|---|
| 4A | **CPU MODULE** | **4A-1** |
| | 4A-1. INTRODUCTION | 4A-1 |
| | 4A-4. LOCAL MEMORY | 4A-1 |
| | 4A-7. THE SERIAL PORTS | 4A-1 |
| | 4A-9. Grounding | 4A-2 |
| | 4A-12. Signal Characteristics | 4A-3 |
| | 4A-16. Parameters | 4A-3 |
| | 4A-19. External Terminal | 4A-3 |
| 4B | **MAIN MEMORY MODULE** | **4B-1** |
| | 4B-1. INTRODUCTION | 4B-1 |
| | 4B-3. MEMORY ALLOCATION | 4B-1 |
| | 4B-6. Program Size | 4B-1 |
| | 4B-8. Data Overflow | 4B-1 |
| 4C | **VIDEO INTERFACE AND KEYBOARD MODULE** | **4C-1** |
| | 4C-1. INTRODUCTION | 4C-1 |
| | 4C-5. ASCII 3.4 - 1968 | 4C-1 |
| | 4C-9. ANSI X3.64 - 1977 | 4C-3 |
| | 4C-11. Cursor Controls | 4C-4 |
| | 4C-14. Enhancements | 4C-4 |
| | 4C-18. Modes | 4C-5 |
| | 4C-25. Erasing | 4C-6 |
| | 4C-27. Cursor Status | 4C-6 |
| | 4C-29. External Line Control | 4C-6 |
| | 4C-31. THE TOUCH SENSITIVE DISPLAY | 4C-7 |
| | 4C-36. THE PROGRAMMER KEYBOARD | 4C-7 |
| | 4C-39. CTRL/SHIFT Keys | 4C-9 |
| | 4C-41. COMPOSITE VIDEO OUTPUT | 4C-10 |
| 4D | **FLOPPY DISK AND INTERFACE** | **4D-1** |
| | 4D-1. INTRODUCTION | 4D-1 |
| | 4D-4. DATA ORGANIZATION | 4D-1 |
| | 4D-10. FILE ACCESS TIME | 4D-2 |
| | 4D-14. FLOPPY DISKS | 4D-2 |
| | 4D-18. THE REAL-TIME CLOCK | 4D-3 |
| 4E | **OPTIONAL ELECTRONIC DISK MODULES** | **4E-1** |
| | 4E-1. INTRODUCTION | 4E-1 |
| | 4E-3. DATA ORGANIZATION | 4E-1 |
| | 4E-9. FILE ACCESS TIME | 4E-1 |
| | 4E-11. BATTERY BACK-UP | 4E-2 |
| | 4E-14. SWITCH SETTINGS | 4E-2 |
| 4F | **DUAL IEEE-488 INTERFACE MODULE** | **4F-1** |
| | 4F-1. INTRODUCTION | 4F-1 |
| | 4F-3. FEATURES | 4F-1 |
| 4G | **REMOTE INTERFACE** | **4G-1** |
| | 4G-1. INTRODUCTION | 4G-1 |

# List of Tables


| TABLE | TITLE | PAGE |
|---|---|---|
| 2-1. | Power Supply Inputs | 2-3 |
| 2-2. | Front and Rear Panel Controls, Indicators, and Connectors | 2-6 |
| 2-3. | Programmer Keyboard | 2-7 |
| 2-4. | Disk Handling Precautions | 2-9 |
| 2-5. | Self-Test Errors | 2-11 |
| 2-6. | Compatible Versions of System Software | 2-12 |
| 3A-1. | Device Names | 3A-2 |
| 3A-2. | Control Codes Available for FUP Generated Command Files | 3A-8 |
| 3A-3. | Control Code Summary | 3A-9 |
| 3D-1. | SET Error Messages | 3D-8 |
| 3D-2. | SET Commands and Arguments | 3D-9 |
| 3E-1. | FUP Error Messages | 3E-13 |
| 3E-2. | FUP Command Options | 3E-14 |
| 4A-1. | RS-232-C Serial Data Ports (KB1: and KB2:) | 4A-2 |
| 4A-2. | CPU S1 Settings | 4A-4 |
| 4C-1. | ASCII Display Chart | 4C-2 |
| 4C-2. | ANSI Control Sequences | 4C-3 |
| 4C-3. | Graphics Mode Characters | 4C-6 |
| 4C-4. | External LED Lines | 4C-7 |
| 4C-5. | CTRL/SHIFT Keys | 4C-9 |
| 4F-1. | ASCII/IEEE-488 - 1978 Bus Codes | 4F-2 |
| 4F-2. | IEEE-488 - 1978 Instrument Ports (Port 0 and Port 1) | 4F-3 |
| 4F-3. | ASCII and IEEE-488 Mnemonic Abbreviations | 4F-4 |
| 4G-1. | Remote Interface Connector (Rear Power Supply) | 4G-1 |

# List of Illustrations


| FIGURE | TITLE | PAGE |
|---|---|---|
| Frontispiece | 1720A Instrument Controller | viii |
| 1-1. | 1720A Manual Set | 1-2 |
| 2-1. | Power Voltage Range Selection | 2-4 |
| 2-2. | Front and Rear Panel Controls, Indicators, and Connectors | 2-5 |
| 2-3. | Programmer Keyboard | 2-7 |
| 2-4. | Inserting the Disk | 2-9 |
| 3A-1. | Start Sequence | 3A-5 |
| 3A-2. | Command File Generator Program | 3A-9 |
| 4-1. | Functional View of the 1720A Instrument Controller | 4-2 |
| 4B-1. | Memory Allocation | 4B-2 |
| 4C-1. | Character Display Program | 4C-3 |
| 4C-2. | Graphics Character Display Program | 4C-5 |
| 4C-3. | Display Worksheet | 4C-8 |
| 4D-1. | Floppy Disk | 4D-2 |
| 4E-1. | E-Disk Select Switch Position | 4E-2 |

**1720A Instrument Controller**

# Section 1
# Introduction

## 1-1. DESCRIPTION OF 1720A DOCUMENTATION

1-2. Several documents are available to serve the needs of a variety of users. Figure 1-1 illustrates the manual set. To obtain additional copies of these, or other manuals when developed, check with a local Fluke Sales Office, listed at the end of this manual. A summary description of each manual is given below:

| | |
|---|---|
| 1720A User Manual | An introductory manual for the programmer or system designer who is using the 1720A to set up an instrumentation system. It presents both the purpose of and the interaction between the various software and hardware resources in the 1720A. In addition, it serves as a reference guide for FDOS, the Console Monitor, and the utility programs. Fluke Part Number 518654. |
| 1720A Fluke BASIC Programming Manual | A description of the Fluke-enhanced ANSI-standard BASIC language developed for the 1720A Instrument Controller. It is arranged by subject, and combines syntax diagrams with examples. Emphasis is on instrumentation system control. This manual presumes familiarity with the 1720A User Manual. Fluke Part Number 518670. |
| 1720A BASIC Reference Guide | A pocket-size quick reference guide that summarizes the most often used contents of the User and BASIC Programming manuals. Fluke Part Number 526210. |
| HPL to Fluke BASIC Handbook | A specialized manual for the programmer who is familiar with the Hewlett-Packard 9825 Calculator using the HPL language. Building upon this familiarity, the handbook describes functional similarities and differences and leads the programmer through the task of converting existing HPL programs for use with the 1720A Instrument Controller. Fluke Part Number 546341. |

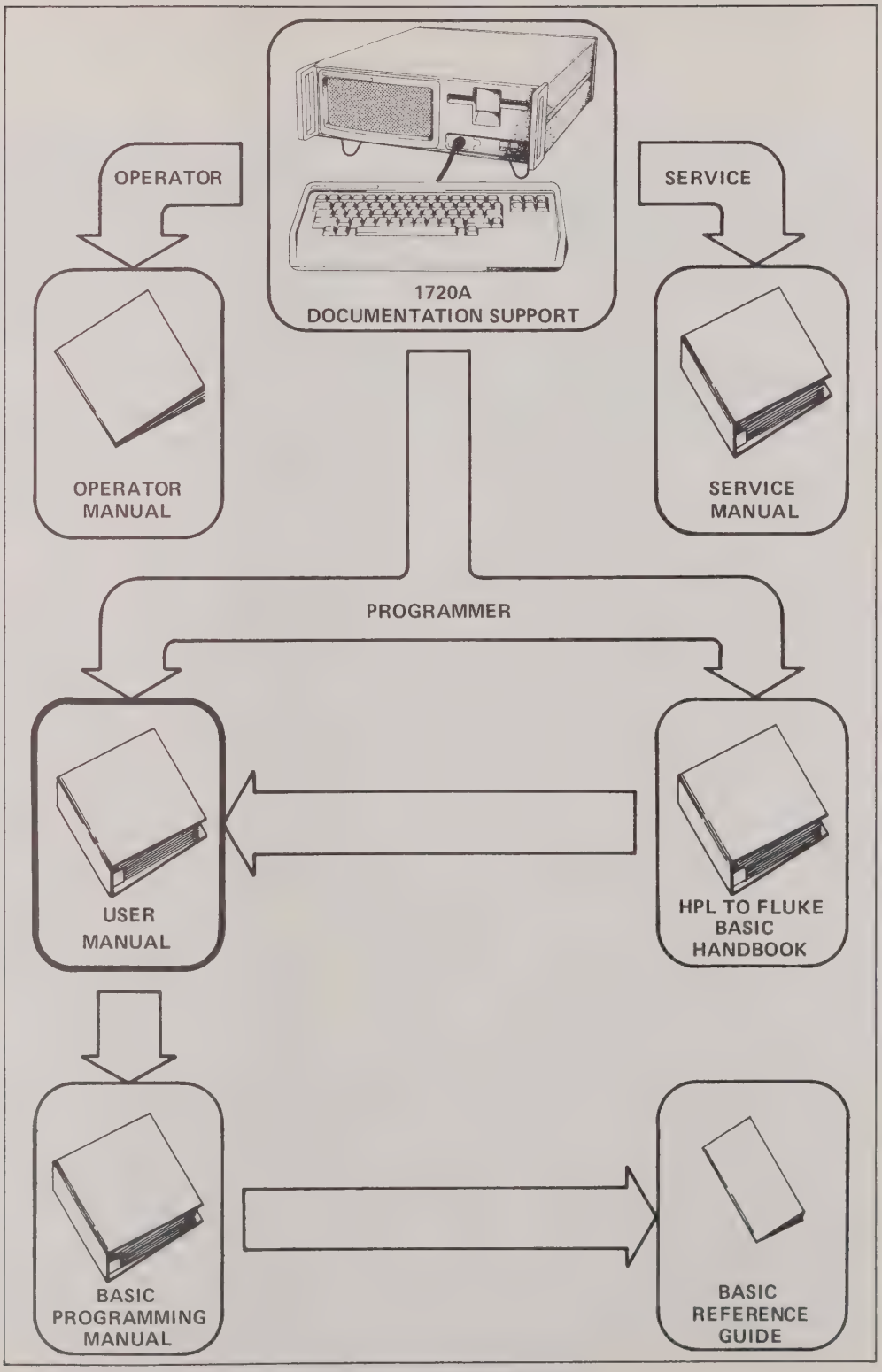



**Figure 1-1. 1720A Manual Set**

| | |
|---|---|
| 1720A Display Worksheet Pads | A grid-pattern worksheet that shows both the normal and double size character positions and the touch sensitive areas. This aids the programmer in the design of effective display layouts. Available in pads of 50. Fluke Part Number 533547. |
| 1720A Operator Manual | A brief manual for the operator of a programmed 1720A-controlled instrument system. It discusses proper disk handling and error interpretation and includes an error and trouble-incident log. Fluke Part Number 518647. |
| 1720A Service Manual | A component level theory of operation for each hardware module, with diagnostic procedures to resolve failures to the modular level. Schematics and parts lists are included. Fluke Part Number 518662. |

## 1-3. INTRODUCTION TO THIS MANUAL

1-4. This manual introduces the 1720A and is a reference for FDOS, the Console Monitor, and the other utility programs provided. It establishes familiarity with both software and hardware capabilities. The Fluke BASIC Programming Manual and other language manuals then build upon this system familiarity describing the use of a programming language and its associated software tools. The subject matter is divided into sections as follows:

| | |
|---|---|
| Section 1 | Introduction. Describes the documentation set, the User Manual, syntax diagrams, and the 1720A Instrument Controller. |
| Section 2 | Installation. Covers unpacking and set-up. Identifies controls, indicators and connectors. Includes start-up procedures and a software module compatibility check. |
| Section 3 | Software. Describes use of FDOS, the Console Monitor, and each system-level utility program. Divided into modular subsections. |
| Section 4 | Hardware. Describes use of each of the hardware modules as a system resource. Divided into modular subsections. |

1-5. Programming languages are covered in separate manuals that presume familiarity with this user manual. Where programming examples are required, Fluke Enhanced BASIC is used unless otherwise specified.

### 1-6. Command Definitions

1-7. The length of any command line cannot exceed 80 characters. Commands are introduced with a syntax diagram that defines legitimate command constructs. A complete definition of the command follows the syntax diagram. The short indented paragraphs (with a "bullet" in place of a paragraph number) following the syntax diagram are part of the definition.

## 1-8. Syntax Diagrams

1-9. Syntax diagrams are used throughout this manual to define correct spelling, punctuation, sequences of words, symbols, and expressions for system and utility commands. The following guidelines define proper use of these diagrams:

- Any path through a diagram starting from the left that does not run contrary to an arrowhead forms a legitimate command construct. The text accompanying the diagram explains legal usage.
- Boldface words in a circular enclosure are to be entered exactly as shown. Example:

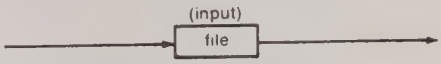
- Key entries with names, such as ESC or RETURN, are shown in a box with rounded corners. Example:

- A required space character entry is always shown as:

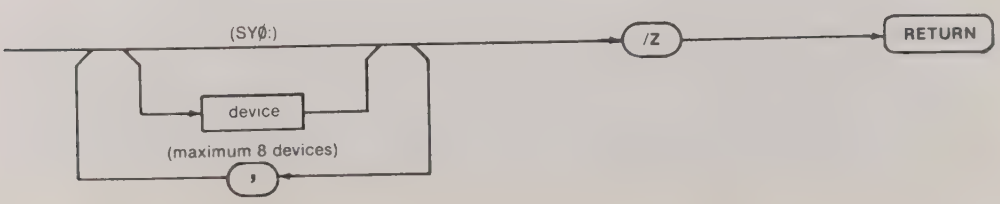
- Control character entries are shown in circular enclosures within angle brackets. The representation CTRL/ means to hold the CTRL key depressed while typing the letter that follows. Example:

- Lower case words enclosed in a box represent other information to be supplied. Example:

- Words outside the path of the diagram, usually in parentheses, provide supplementary information. These words are normally not part of the definition of the statement. Example:




1-10. The following syntax diagram of the zero directory command of the File Utility Program illustrates these points. The diagram shows that the basic construction of the command is /Z. The supplementary note shows that the default system device (SY0:) is assumed when one is not specified. The diagram also shows that one or more devices may be specified, separated by commas.



## 1-11. THE 1720A INSTRUMENT CONTROLLER

1-12. The Fluke 1720A Instrument Controller was designed for the task of building and controlling instrument systems. Because it has a removable Programmer Keyboard, a Touch-Sensitive Display for operator interaction, and a rack-mountable package, it can be used on the programmer's desk or rack mounted in an industrial instrumentation system.

1-13. The 1720A includes a 16-bit microcomputer and is supplied with a System Disk containing Fluke-designed system software, utility programs, and one or more programming languages. Dual IEEE-488-1978 standard ports allow the 1720A to control up to 28 compatible instruments or peripheral devices. In addition, two RS-232-C ports permit the 1720A to interface with a wide variety of general purpose peripherals and data communications equipment.

1-14. The 1720A is not a business computer or calculator that has been adapted to the job of an industrial controller. Instead, the 1720A is a special purpose machine that has a variety of standard features specifically designed for controlling industrial instruments. These standard features include:

- A rack-mount package that occupies a minimum of valuable rack space.
- A full-width touch-sensitive display that allows the design of operator control interfaces that are flexible, simple to use, and specifically adapted to the process.
- A detachable, ASCII keyboard that allows separation of programming functions from daily operation.
- A two-port IEEE-488-1978 interface that allows the 1720A to function as two controllers with up to 14 instruments in each system.
- A software development system that includes an operating system, a monitor, utility programs, and one or more programming languages with an editor and other supporting software tools.
- A double-density floppy disk drive that allows storage of programs and files on a permanent media. The floppy disks provided with the 1720A contain the standard software development programs as well as a complete set of diagnostic software for isolation of failures to the module level.
- A soft-loaded software architecture that allows field upgrading of software as new capabilities and languages are developed.
- A 60K byte memory module that provides the main memory area for program execution, file transfer, and the execution area for all system software.
- An optional electronic disk memory module, E-Disk™, that provides an additional 128K or 256K bytes of storage. It is intended for use as a fast access operation media. Internal battery backup can sustain the contents of the electronic disk for a short time when AC power is removed. This backup time may be extended indefinitely by using an external battery. (E-Disk is a trademark of the John Fluke Mfg. Co., Inc.)

- Two independent software-controlled RS-232-C ports that provide serial access to a wide variety of general-purpose peripherals and data communications equipment. Software control of the ports eliminates mechanical switches for baud rate selection, and gives flexibility in protocol.

- A real-time clock that may be used as a determining program variable during program execution. It also allows application programs to date- and time- stamp data files as they are collected.

# Section 2
# Installation and Start-Up

## 2-1. INTRODUCTION

2-2. This section covers unpacking and installation. In addition, it provides a description of the controls, indicators, and connectors. Following this, it presents start-up procedures for putting the 1720A into operation.

## 2-3. UNPACKING

2-4. The 1720A Instrument Controller is carefully packed for shipping to ensure that the unit arrives in good condition. Unpack all containers carefully and check all packing materials for accessories, cables, and manuals. Do not dispose of the packing materials before inspecting for shipping damage. If this inspection reveals damage or indicates that damage might have occured, notify the shipper immediately. Then call a Fluke Sales Office or Customer Service Office. A checklist of the standard and optional items included with the 1720A Instrument Controller is presented in paragraph 2-6.

2-5. If the 1720A Instrument Controller needs to be shipped again at a later date, the original packing carton should be used with all fillers properly in place. Fluke does not recommend shipping the 1720A in a substitute container. To obtain an approved shipping container, call any Fluke Sales Office.

## 2-6. Unpacking Checklist

1. Controller Mainframe
2. Programmer Keyboard
3. Power Cord
4. System Disk
5. Diagnostics Disk
6. Data Disk
7. This User Manual
8. 1720A Fluke-Enhanced BASIC Programming Manual
9. 1720A Fluke BASIC Reference Guide (pocket-size)
10. 1720A Operator Manual
11. 1720A Service Manual
12. 1720A Controller Programming Work Sheet (Pad of 50)

## 2-7. Options and Accessories

2-8. These items are not included in the shipment unless ordered separately. They may or may not be separately packaged.

1. 128K Byte Electronic Disk Module, Option 1720A-001, maximum two per unit. These will be installed in the mainframe if ordered with the Instrument Controller.

2. Extra Programmer Keyboard (accessory Y1720).

3. Ten-pack of blank floppy disks, unformatted (accessory Y1706).

4. IEEE-488-1978 Cables, one-meter length (accessory Y8001), two-meter length (accessory Y8002), four-meter length (accessory Y8003).

5. RS-232-C Standard Interface Cable, allows connection to standard telephone data sets and modem-type devices. Two-meter length (accessory Y1707). Ten-meter length (accessory Y1708).

6. RS-232-C Null Modem Cable, 0.3 meter length. This cable allows connection to standard data terminal devices (accessory Y1705).

7. RS-232-C Printer Cable, two-meter length. This cable is designed for connecting the 1720A to a Fluke 1776A Serial Printer (accessory Y1709).

8. Rack Mounting Kit, includes 61 cm (24 inch) slides (accessory number Y1790).

9. Shipping Case (accessory Y1711).

10. Customer Maintenance Kit, includes the tools and accessories that are required by the diagnostic and troubleshooting procedures in the 1720A Service Manual. This kit is included in the Module Service Kits below. Order through Fluke Service Centers as part number 577791.

11. Module Service Kit, includes one of each of the functional modules, the Maintenance Kit of required tools, and a selection of additional parts and tools. This is the basic spares kit for modular servicing. Order through Fluke Service Centers as part number 539379.

12. Module Service Kit with E-Disk™, includes one of each of the functional modules, the Maintenance Kit of required tools and a selection of additional parts and tools. This kit is identical to the 539379 kit except for the addition of an E-Disk™ module and associated spares. Order through Fluke Service Centers as part number 537506.

13. CRT Display Service Kit, includes the CRT display tube with factory aligned yoke, the high voltage transformer, and the horizontal and vertical drive electronics. Order through Fluke Service Centers as part number 537472.

14. 70% Level Service Kit, includes a minimum number of functional modules that have been calculated or shown by experience to have the lowest mean time between failure (MTBF). This kit must be supplemented by a Maintenance Kit. Order through Fluke Service Centers as part number 539361.

## 2-9. INSTALLATION

2-10. Installation of the 1720A requires locating the 1720A on a suitable table, attaching the keyboard and line cord, and turning on the power. The Power Switch is located on the rear panel. The line voltage has been factory set as ordered. This voltage setting is indicated on the rear panel decal.

### 2-11. Line Voltage Range Selection

2-12. The 1720A Instrument Controller uses a high-efficiency switching power supply that will operate from any of the inputs shown in Table 2-1.

**Table 2-1. Power Supply Inputs**

| VOTAGE RANGE | LINE FREQUENCY | SWITCH SETTING | FUSE |
|---|---|---|---|
| 90V to 132V | 47 Hz to 63 Hz | 115 | 3A SB |
| 192V to 250V | 47 Hz to 63 Hz | 230 | 1.5A SB |
| 104V to 126V | 380 Hz to 420 Hz | 115 | 3A SB |

2-13. If line voltage requirements change, it will be necessary to set the power supply line voltage switches. It may also be necessary to change the fuse to match the power source and install an appropriate power plug.

**CAUTION**

**Do not operate the 1720A from a 230V-range source with the 3 amp fuse installed. This size fuse may be unable to protect the unit from internal damage if a failure-caused overload occurs.**

**WARNING**

**TO AVOID ELECTRICAL SHOCK HAZARD, REMOVE THE POWER CORD FROM THE 1720A AT LEAST FIVE (5) MINUTES PRIOR TO PERFORMING THE FOLLOWING PROCEDURE. THIS WILL ALLOW TIME FOR THE INTERNAL CAPACITORS TO DISCHARGE TO A SAFE LEVEL.**

2-14. To gain access to the voltage range setting, use a #1 Phillips screwdriver to remove the three screws holding the Power Supply Module to the chassis and slide the power supply straight out the rear. Facing the rear of the unit, the power supply is the right half-panel. The two slide switches are located under the right side of the top circuit board as illustrated in Figure 2-1, and are labeled with the numbers 115 and 230.

2-15. The fuse is accessible without removing the power supply, and is on the right side of the rear panel. Be sure that the fuse value corresponds to the setting of the voltage range switches as given in Table 2-1.

### 2-16. Environment

2-17. The 1720A requires adequate ventilation to maintain the correct operating temperature. Keep the rear panel at least 10 cm. (4 inches) from the wall so that the fan can be effective. There is no minimum distance for top or bottom placement, as the controller is designed to be mounted in a rack with other instruments.

2-18. The 1720A operates reliably within the temperature range of 10°C to 40°C (50°F to 104°F). The controller may be turned on at lower temperatures, down to -10°C (14°F), to help warm it up to room temperature. The relative humidity should be within the range of 8% to 80%, non-condensing.

**CAUTION**

**Low humidity environments can induce electrostatic discharges within the rotating floppy disk, and between the touch-sensitive display and the operator. Such discharges can permanently damage high-density MOS microcircuitry within the 1720A.**

2-19. The 1720A must be stored (non-operating) within the temperature range of –10°C to 60°C (14°F to 140°F), and within a humidity range of 5% to 95%, non-condensing.

2-20. Floppy disks are more sensitive to storage environments than the instrument controller. They must not be allowed to become colder than 10°C (50°F), or warmer than 52°C (126°F). Relative humidity must be kept within the range 8% to 90%, non-condensing. If a disk has been stored outside these limits, the best way to attempt recovery of data is to allow it to slowly reach room temperature and humidity at least 3 to 4 hours before placing it in the controller. The Start-up Procedures include other disk handling precautions.

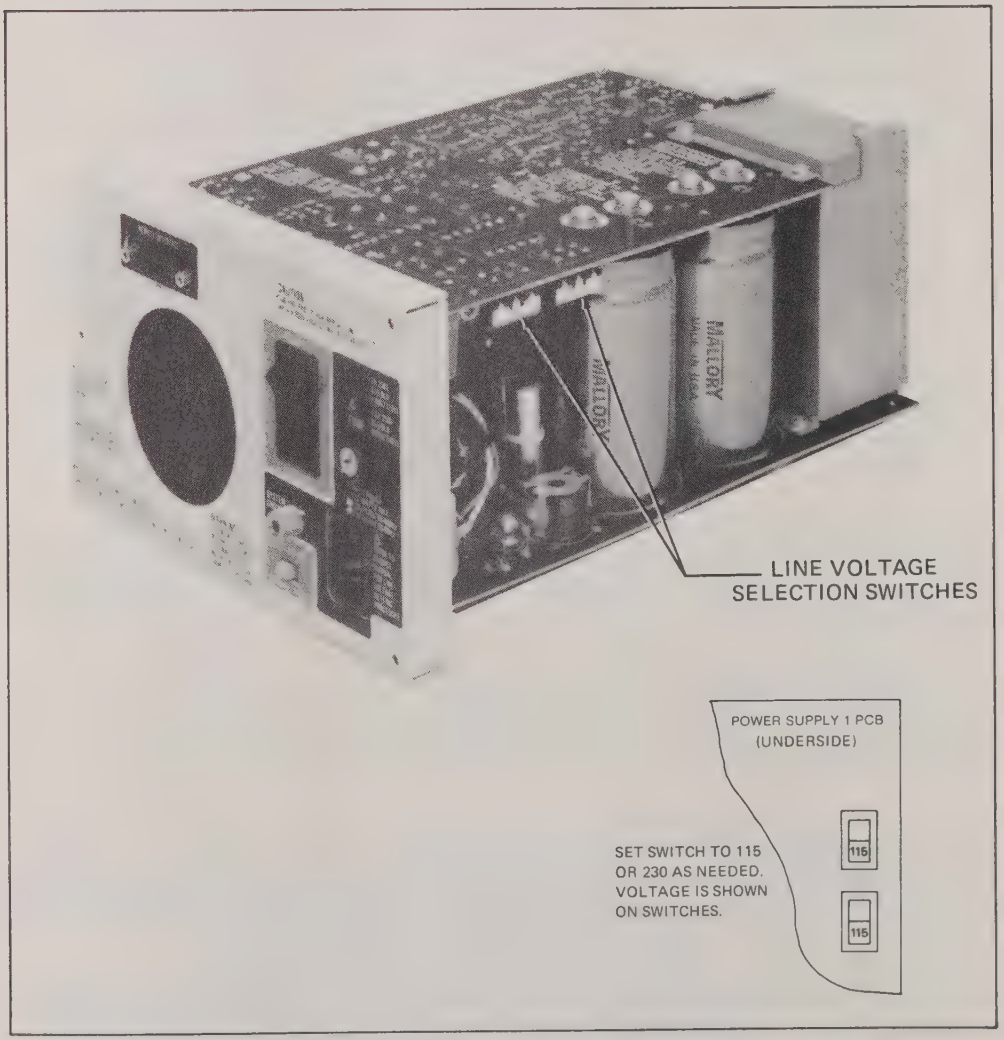



**Figure 2-1. Power Voltage Range Selection**

## 2-21. CONTROLS, INDICATORS, AND CONNECTORS

2-22. Figure 2-2 and Table 2-2, illustrate and describe the front and rear panels, and Figure 2-3 and Table 2-3, illustrate and describe the programmer keyboard.

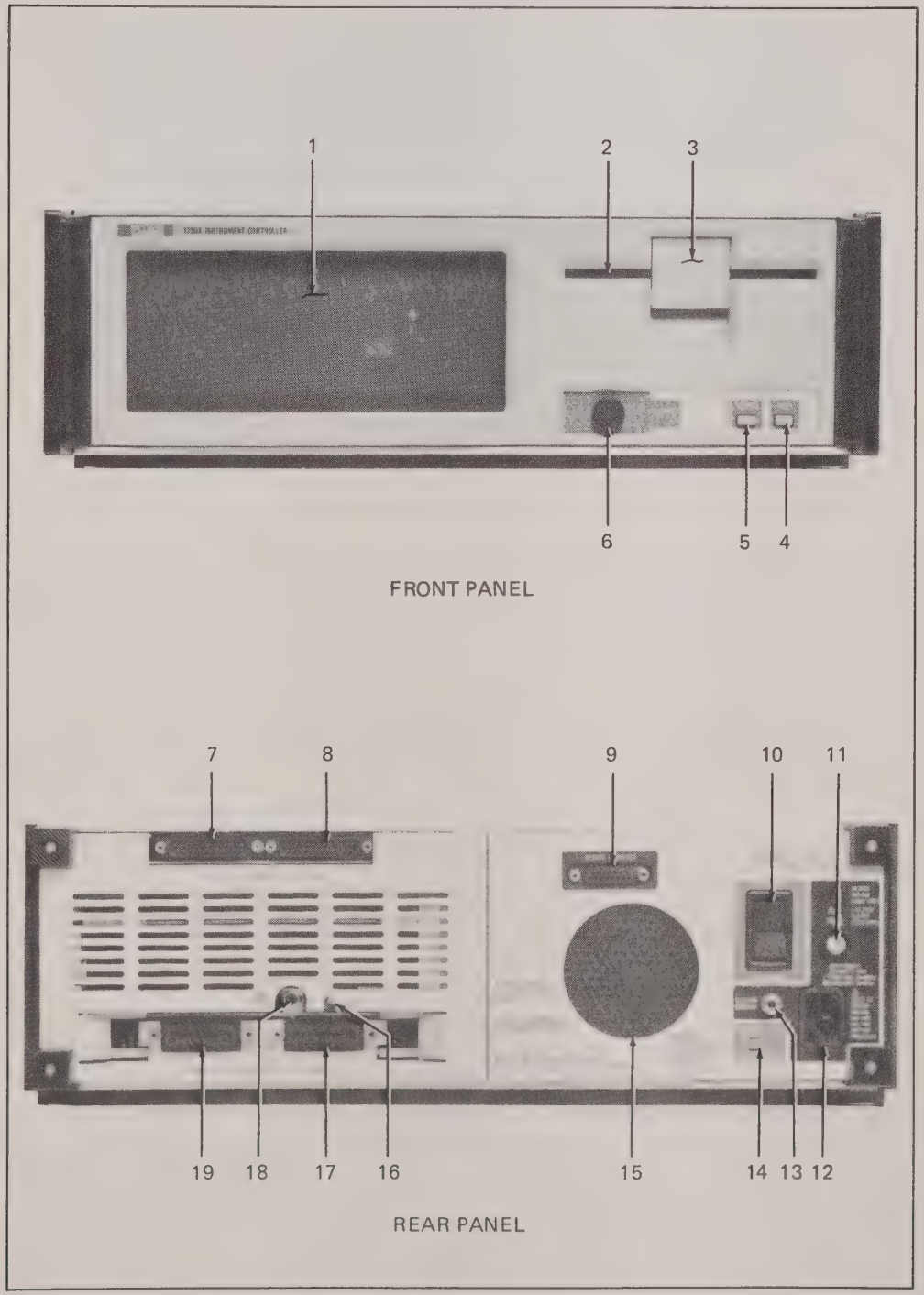



**Figure 2-2. Front and Rear Panel Controls, Indicators, and Connectors**

**Table 2-2. Front and Rear Panel Controls, Indicators, and Connectors**

| REF. NO. | NAME | FUNCTION |
|---|---|---|
| 1 | Touch-Sensitive Display | A display with touch sensitive areas which allows operation of a program by touch according to commands shown on the screen. The commands are program controlled and generated. |
| 2 | Disk Loading Slot | The floppy disk is loaded into the disk loading slot for program execution and data storage. |
| 3 | Disk Entry Latch | Seats disk for operation. |
| 4 | ABORT Push Button Switch | Clears devices on the IEEE-488 buses and either stops the program or allows automatic program continuation, depending on the application program. |
| 5 | RESTART Push Button Switch | Initiates self test, clears main memory, and reloads program. |
| 6 | Keyboard Connector | For connecting the detachable keyboard. |
| 7 | RS-232-C Connector #1 | Standard I/O port for peripheral devices. (KB1:) |
| 8 | RS-232-C Connector #2 | Standard I/O port for peripheral devices. (KB2:) |
| 9 | Remote Interface Connector | Provides for external E-Disk battery back-up, external control of ABORT and RESTART switches, and two software controlled output lines. |
| 10 | Main Power Switch | Applies and removes line power to the instrument controller. |
| 11 | Fuse | Power supply line fuse. |
| 12 | Power Connector | Input line power connector. |
| 13 | System Ground | A common ground for system components. |
| 14 | E-Disk Enable Switch | Enables battery backup for E-Disk. |
| 15 | Air Filter | Filters air used to cool power supply. See maintenance procedures. |
| 16 | Display Intensity Control | A screwdriver adjustment which controls the intensity of the display. |
| 17 | IEEE-488 Port 1 Connector | Standard IEEE-488-1978 connector for instrumentation. |
| 18 | Composite Video Connector | Drives standard data display monitor. |
| 19 | IEEE-488 Port 0 Connector | Standard IEEE-488-1978 connector for instrumentation. |

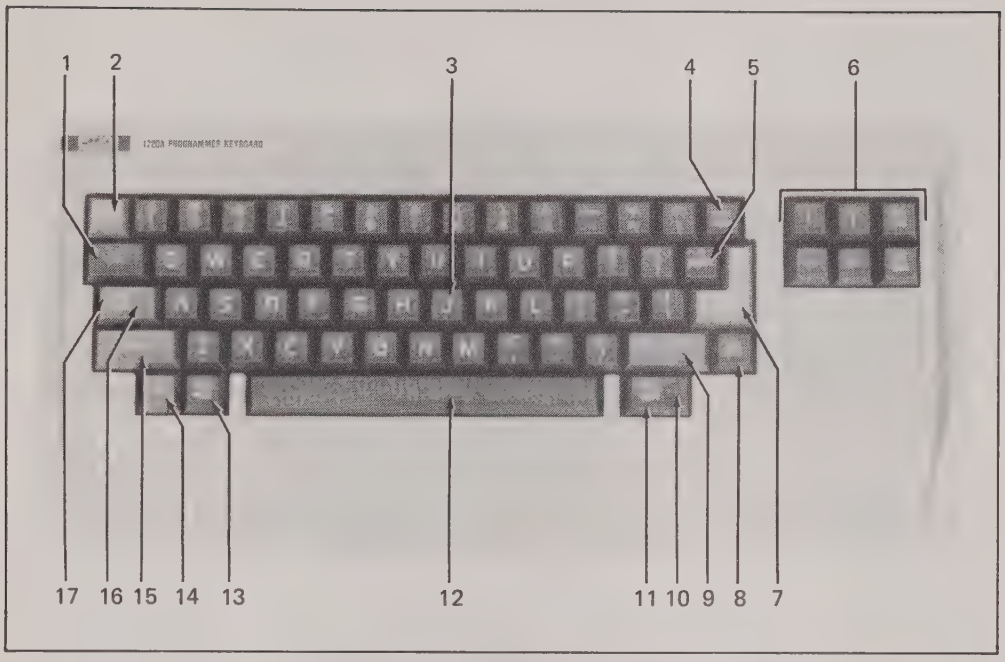



**Figure 2-3. Programmer Keyboard**

**Table 2-3. Programmer Keyboard**

| REF. NO. | NAME | FUNCTION |
|---|---|---|
| 1 | TAB Key | Not used at this time (Horizontal Tab code). |
| 2 | ESCape Key | Escape code. |
| 3 | Character Entry Keys | In standard typewriter layout, for entry of characters and numbers into the 1720A. |
| 4 | BACK SPACE Key | Functional only in Edit Mode. |
| 5 | DELETE Key | Delete the character just entered and backspace to the previous position. |
| 6 | Editing Keypad | Functional only in EDIT Mode. Arrows indicate direction of cursor movement. DELete LINE and DELete CHARacter act upon the current cursor position. |
| 7 | RETURN Key | General-purpose entry key. Command or statement entries in all modes are stored in temporary buffer unit RETURN is pressed. The display then moves the cursor to the start of the next line. |
| 8 | LINE FEED Key | Functional only in Edit Mode. |
| 9 | SHIFT Key | A key modifier that must be held down while another key is pressed to select the upper character of any key that has two characters on it, or the CAPITAL of any alpha character. |

Table 2-3. Programmer Keyboard (cont)

| REF. NO. | NAME | FUNCTION |
|---|---|---|
| 10 | Page Mode Indicator | ON when Page Mode has been selected. See Reference Number 11. |
| 11 | PAGE MODE | Alternate action switch that selects or cancels Page Mode. In Page Mode, the indicator is lighted, and display output stops at the bottom without scrolling. See Reference Number 13. |
| 12 | Space Bar | Generates space characters. |
| 13 | NEXT PAGE Key | Functional in Page Mode only. Clears the screen and allows an additional 16 display lines. |
| 14 | CTRL (Control) Key | A key modifier that must be held down while another key is pressed. |
| 15 | SHIFT Key | A key modifier that must be held down while another key is pressed to select the upper character of any key that has two characters on it, or the CAPITAL of any alpha character. |
| 16 | Caps Lock Indicator | ON when Caps Lock Mode has been selected. See Reference Number 17. |
| 17 | CAPS LOCK | Alternate action switch that places the controller into and out of Caps Lock Mode. In Caps Lock Mode the indicator is lighted, and all alpha characters are automatically capitalized. Other characters remain unchanged. |

## 2-23. START-UP PROCEDURE

2-24. Following is a step by step procedure for starting up the 1720A Instrument Controller. This procedure is for the standard 1720A, supplied with Fluke-enhanced BASIC.

1. Check for proper line voltage selection as discussed under Installation earlier in this section. Plug the power cord into a properly grounded outlet.

**WARNING**

**SAFE OPERATION OF THE 1720A, AS WELL AS RELIABLE SYSTEM OPERATION REQUIRES A GROUNDED POWER CONNECTION.**

2. Apply power to the system by depressing the top of the power switch located on the rear panel.

3. Release the door latch on the floppy disk drive by pressing in on the top of the door, as shown in Figure 2-4.

4. Read the disk handling precautions in Table 2-4. Gently insert the System Disk into the disk drive with the label upward, as shown in Figure 2-4, until the disk is fully seated. Then latch the drive closed by pressing the bottom of the door.

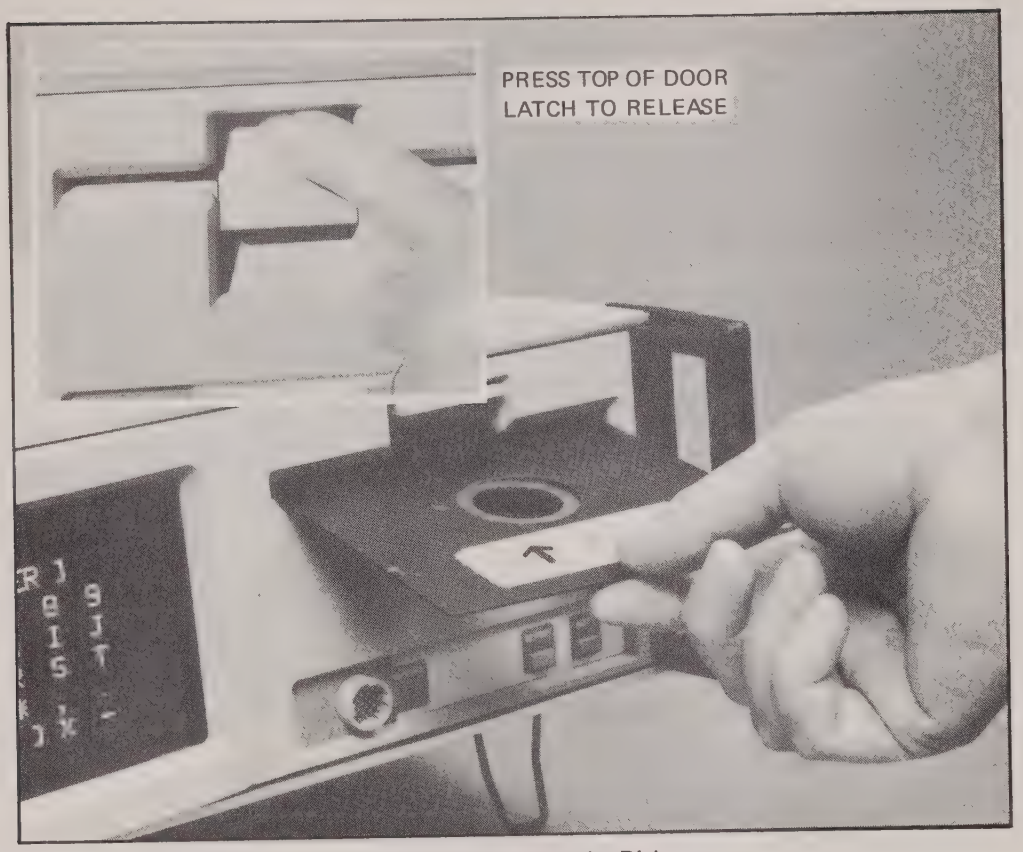



**Figure 2-4. Inserting the Disk**

**Table 2-4. Disk Handling Precautions**

| | |
|---|---|
| 1. | Keep the disk in its protective folder when not in use. |
| 2. | The disk surface can be easily damaged. Do not touch the exposed disk surface with hands or any material which may damage it. |
| 3. | Careless handling may damage the disk. Avoid dropping, throwing, or twisting the disk. |
| 4. | Store disks vertically and support them so they will not bend. Stacking may distort them and affect their contents. Avoid placing heavy objects on the disks. |
| 5. | Exposure to heat and sunlight may warp the disks. |
| 6. | Magnetic sources may erase data on the disks. Keep the disks away from electric motors, generators, and transformers. |
| 7. | Mark disk labels before placing them on the disk envelope or use a felt tip pen. Pressure from a ball point pen or pencil may destroy data on the disk. |
| 8. | Temperature affects the performance of the disk. Do not use a disk that is overheated or overcooled until it has reached room temperature. |

5. Press RESTART. This step may be avoided by inserting the disk before turning power ON; however, some of the following messages would not appear as the display system warms up to normal intensity.

6. Observe the display. It should read:

```
FLUKE 1720A CONTROLLER
HELLO
BOOT V1.2
```

7. Verify that the version number is 1.2. If it is not, check the front of this manual for supplemental information, or call a Fluke Customer Service Center for advice.

8. Following this display, the 1720A then performs a self-test procedure from permanent (ROM) memory located on the CPU module. The display changes the `BOOT V1.2` to `SELF TEST`. The Self Test functionally tests the following modules:

a. CPU Module

b. Floppy Disk Interface

c. Memory Module

d. Electronic Disk (if present)

e. Keyboard/Video Interface

f. IEEE-488 Interface

9. Refer to Table 2-5 if any error messages occur. This table presents the error messages that the self-test program can generate.

10. After the self-test, the 1720A loads the Floppy Disk Operating System software (FDOS) into main memory from the floppy disk. During the loading process, the display reads:

```
FLUKE 1720A CONTROLLER
LOADING
```

11. When loading is complete, FDOS takes control and attempts to load a System Command File from the disk by looking for a file named COMMND.SYS. A command file contains stored keyboard commands that would otherwise have to be typed in. Section 3 explains these concepts.

a. If the file COMMND.SYS is not found, FDOS loads the Console Monitor (CONMON) directly and transfers control to it. This may be the case if the disk being loaded is not the system disk supplied with the 1720A, or the COMMND.SYS file has been renamed or deleted. The Console Monitor displays a # prompt character. Section 3 explains the use of CONMON.

b. If the file COMMND.SYS is found, it is loaded and command file status is designated active. The display then reads:

```
Command File Active
```

12. The next action depends on the contents of the command file. The System Disk supplied with the 1720A contains a short system command file that first checks the TIME utility to see if the time has been set. It then loads the BASIC interpreter and transfers control to it.

**Table 2-5. Self Test Errors**

| MESSAGE | ACTION |
|---|---|
| !Memory Error | Indicates a problem with the memory board. Service may be required. |
| !Video Error | Indicates a problem with the video board. Service may be required. |
| !IEEE Missing or Faulty | Indicates a problem with the IEEE Bus Interface board. Check that all bus instruments connected to the controller are ON and that all cables are securely in place. If everything seems correct and the message still appears when RESTART is again pressed, service may be required.<br>*NOTE*<br>*Disconnect the IEEE Bus cables from the controller to confirm that the instruments are the source of trouble.* |
| !Floppy Error | Indicates a problem with the floppy disk drive or its control board. Service may be required. |
| ?No System on Device | Check that the disk in the drive is a system disk. Try another system disk. The wrong disk may be inserted or it may be inserted incorrectly. |
| ?Disk not Mounted | Insert or reinsert a system disk. Either there is no disk inserted or it is improperly inserted. Check to see if the disk is upside down or the entry latch has not been closed. |
| ?Floppy Error | The inserted disk is faulty. Try another system disk. |
| ?Bad Directory on Device | The directory on the disk is incorrect. Use another disk. |

13. If you are using the System Disk, and the time clock has not previously been set, the display will next read:

```
Enter Date:  DD-MM-YY
```

14. Type in today's date in numeric form, starting with the day, then the month, and then the year. The entries must be separated by any non-numeric character. Use DELETE to correct any mistakes. Press RETURN and the display reads:

```
Enter Time:  HH-MM
```

15. Type in the time, in 24-hour form, starting with the hour, followed by the minutes. The entries must be separated by any non-numeric character. Press RETURN and the display reads:

```
BASIC  (time)  (date)

Basic Version 1.2

Ready
```

*NOTE*

*The 24-hour day starts at 00:00 (midnight) and ends at 23:59 (just before midnight). If the time is at or beyond midnight and before 1:00 A.M., subtract 12 hours. If the time is at or beyond 1:00 P.M., and before midnight, add 12 hours. Otherwise, use the time unchanged.*

Three examples:

| TIME | 24-HOUR TIME |
|---|---|
| 12:30 A.M. | 00:30 |
| 9:45 A.M. | 09:45 |
| 4:20 P.M. | 16:20 |

16. The 1720A is now in Immediate Mode BASIC. Check that the version number is 1.2. If it is not, check the front of this manual or the Fluke BASIC Programming Manual for supplemental information, or call a Fluke Customer Service Center for advice. Refer to the 1720A Fluke BASIC Programming Manual for details on the use of BASIC in the 1720A Controller. Information on loading and the use of other languages is presented in separate manuals.

## 2-25. SOFTWARE COMPATIBILITY CHECK

2-26. Software modules are supplied with the 1720A Instrument Controller as files on the System Disk. These machine-language program files are interdependent and are compatible only in the combinations of versions supplied by Fluke and documented in published Fluke manuals, including addendum inserts. The portability and copying ease of Fluke system software allows the user to take advantage of the continuing program of software development at Fluke. The user should make sure however in copying new or updated Fluke system software that all of the software modules are compatible. This can be verified by obtaining the appropriate Fluke manuals and then following the procedure below.

2-27. Because software modules are easily erased and copied, do not assume that the label on a Fluke System Disk is valid unless the disk is new. The use of incompatible system software modules will produce unpredictable results for which Fluke cannot provide analytical, diagnostic, or other software support, except for identification of compatible combinations.

### 2-28. Compatible Versions of System Software

2-29. The versions of Fluke system software modules that are compatible and supported in this manual are listed in table 2-6. If subsequent software releases are also supported by this manual, an addendum insert is available that identifies the compatible versions and any functional changes and additions.

**Table 2-6. Compatible Versions of System Software**

| NAME | VERSION | NAME | VERSION |
|---|---|---|---|
| BOOT | 1.2 | TIME.CIL | 1.0 |
| FDOS.SYS | 1.3 | SET.CIL | 2.0 |
| CONMON.SYS | 1.1 | FUP.CIL | 1.1 |

*NOTE*

*BOOT is contained in PROM memory on the CPU module. All other system software modules are supplied as disk files.*

## 2-30. Procedure

1. With power ON, insert the System Disk and press RESTART.

2. Check that the initial display includes `BOOT V1.2` .

3. If the final display includes `Command File Active` the version of FDOS will not be displayed unless the system command file is temporarily renamed as follows:

   a. Enter ⟨CTRL/P⟩ (Hold CTRL down while typing P.)

   b. After the prompt #, type `F` RETURN .

   c. After the prompt *, type `MFϕ:/A` RETURN .

   d. After the prompt *, type `COMMND.CMD=COMMND.SYS/R` RETURN .

   e. Press RESTART

4. Check that the final display includes `FDOS Version 1.3` and `Console Monitor Version 1.1`.

5. After the prompt #, type `T` RETURN . Check that the resulting display includes `Time Version 1.0`. Then enter ⟨CTRL/P⟩.

6. After the prompt #, type `S` RETURN . Check that the resulting display includes `Set Version 2.0` . Then enter ⟨CTRL/P⟩.

7. After the prompt #, type `F` RETURN . Check that the resulting display includes `File Utility Program Version 1.1`.

8. If the system command file was temporarily renamed in step 3, restore it as follows:
   After the prompt *, type `COMMND.SYS=COMMND.CMD/R` RETURN .

9. Press RESTART.

2-31. If any of the displays do not agree with these software versions, check the front of this manual for supplemental information, or call a Fluke Technical Service Center for advice.

**CAUTION**

**Do not continue to use the 1720A with system software modules that are not known to be compatible and supported by a published Fluke manual or manual addendum. To do so will produce unpredictable results for which Fluke cannot provide analytical, diagnostic, or other support, except for identification of compatible combinations.**

# Section 3
# System Software

## 3-1. INTRODUCTION

3-2. System Software developed for the 1720A Instrument Controller consists of distinct modules that serve specific purposes. This section begins with an overview of the software package that gives a brief description of each software module. The remainder of the section consists of a series of detailed functional descriptions of each software module. Fluke-Enhanced BASIC and other programming languages are covered in separate manuals.

3-3. This section is divided into subsections that separately discuss each of the system software modules. These subsections are summarized below.

## 3-4. FDOS

3-5. The Floppy Disk Operating System (FDOS) performs file management for all programs and responds to interrupts from input and output (I/O) devices, such as the keyboard and the floppy disk. Whenever the 1720A is operating under normal conditions, FDOS is present in Main Memory. FDOS also includes a processor that allows command files of stored user defined keyboard inputs to be processed.

## 3-6. CONMON

3-7. The Console Monitor (CONMON) is a system program that works through FDOS to simplify the process of loading and transferring control to the other software tools available to the programmer. This software, including CONMON, remains on the system disk, or when configured, in the optional electronic disk until needed. This arrangement allows the 1720A to make efficient use of the Main Memory.

## 3-8. TIME

3-9. The Time and Date Set utility (TIME) enables the user to set the time and date into the system clock where it is accurately maintained for use by other programs.

## 3-10. SET

3-11. The Set Serial Port utility (SET) enables the user to independently set the baud rate, number of data bits, stop bits, parity, and line and file terminator characters for each RS-232-C Serial Port.

## 3-12. FUP

3-13. The File Utility Program (FUP) is a file transfer and management program that gives the programmer flexible control over the files on floppy disks and on the optional electronic disk. FUP has many other capabilities which are explained fully.

# Section 3A
# FDOS

## 3A-1. INTRODUCTION

3A-2. The Floppy Disk Operating System (FDOS) is a machine-language system software file that is supplied on the System Disk with the file name FDOS.SYS. When loaded into Main Memory it responds to interrupts from input and output devices, such as the keyboard, the touch-sensitive display, and the floppy disk. In addition, FDOS performs file management on the floppy and electronic disks for all other programs. Whenever the 1720A is operating under normal conditions, FDOS is present in Main Memory. FDOS also includes a processor that allows command files of stored user defined keyboard inputs to be processed automatically when called. This concept is explained in the discussion of command files in this subsection.

## 3A-3. DEVICE NAMES

3A-4. FDOS uses symbolic names to identify the source and destination of any information transfer, except for IEEE-488 bus instruments. Table 3A-1 summarizes these device names.

- Two letters followed by a numeric digit and a colon are required.
- When a file device is specified (MF$\phi$: or ED$\phi$:), the device name is used as a prefix to a file name. Example:

  ```
  MFϕ:TEST.Q44
  ```

- When no device is specified, the default for file operations is the system device (SY$\phi$:). Other defaults are defined individually.
- SY$\phi$: is always redundant and does not need to be specified.
- Usage of device names is identical for any utility program or programming language.
- Other references in this manual: SET and FUP.

## 3A-5. THE SYSTEM DEVICE

3A-6. The System Device is a useful default provided in FDOS for file operations.

- When a file name is specified without a device, the System Device is assumed.

- Normally the System Device is the floppy disk (MFØ:).
- The System Device can be reassigned to the electronic disk by a FUP command. This process can be automated at start-up in a command file.
- Other references in this manual: FUP.

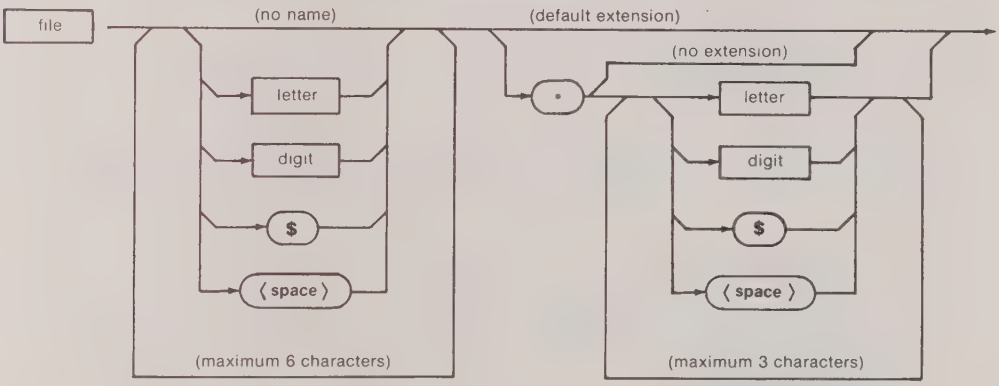
3A-7. The System Device will be automatically defined as the electronic disk at start-up time if the following conditions are met:

1. The file FDOS.SYS is not found on the floppy disk.
2. The electronic disk is installed.
3. The electronic disk contains a valid directory and the file FDOS.SYS.

**Table 3A-1. Device Names**

| | | | |
|---|---|---|---|
| SYØ: | System Device | KBØ: | Keyboard and Display |
| MFØ: | Floppy Disk | KB1: | RS-232 Serial Port 1 |
| EDØ: | Optional Electronic Disk | KB2: | RS-232 Serial Port 2 |
| MMØ: | Main Memory (FUP only) | | |

## 3A-8. FILE NAMES




3A-9. FDOS uses file names to identify programs and data files stored on a file device (floppy or electronic disk).

- A file name is 1 to 6 letters, numeric digits, or $ or space characters, in any combination. Examples:

  ```
  TESTB2   $25795
  ```

- The optional file name extension is a period followed by 1 to 3 letters, numeric digits, or $ or space characters, in any combination. Examples:

  ```
  TEST.T43   FILE9.$3
  ```

- Since a file is uniquely identified by its name, there can be only one occurrence of any file name on each file device.

- When FDOS is unable to identify a file name from the context of an operation, it will search for, or assign, a file with no name. There can be only one such no-name (all-space) file on each file device.

- A period at the end of a file name without an extension specifies a blank (3-space) extension.

- A file name without an extension specifies a default extension that is defined by the utility program or language in use.

- Other references in this manual: FUP.

3A-10. The following file names are used by FDOS to identify special purpose files:

| NAME | MEANING |
|---|---|
| FDOS.SYS | Floppy Disk Operating System |
| CONMON.SYS | Console Monitor |
| TIME.CIL | Time and Date Set Utility |
| SET.CIL | Set Serial Ports Utility |
| FUP.CIL | File Utility Program |
| BASIC.CIL | Fluke-Enhanced BASIC Interpreter |
| COMMND.SYS | System Start-up Command File |
| FUP.HLP | File Utility Program Help Display File |

3A-11. The following file name extensions are used by FDOS to identify certain types of files:

| EXTENSION | MEANING |
|---|---|
| .SYS | System-level programs and command file |
| .CIL | Utility programs (Core Image Load) |
| .BAS | ASCII test BASIC programs |
| .BAL | Lexically analyzed BASIC programs |
| .HLP | System level data files |
| .CMD | Utility command files |

## 3A-12. COMMAND FILES

3A-13. A command file is a sequence of Programmer Keyboard inputs stored in a file. Command files are identified by the form of the file name. When processed, the contents of a command file are treated as if they were input from the Programmer Keyboard, except for the ? and & characters which identify special operations.

3A-14. Two types of command files are available:

1. COMMND.SYS is the system command file. It operates only at start-up time.

2. (File name).CMD is a utility command file. It operates whenever its file name is given as input to the Console Monitor.

3A-15. The following characteristics are common to all command files:

- All 1720A keyboard operations are accessible from a command file.

- Command file processing begins from the Console Monitor.
- Each line contains one keyboard input command.
- Command file lines are processed in sequence only.
- The & character identifies a line to be sent directly to the display. It may include display control sequences to format a message.
- If a display line includes either the & or ? characters, each must be preceded by a backslash ( \ ).
- Only one command file can be active at any time.

3A-16. A command file remains active until one of the following occurs:

1. Its last line has been processed.
2. Control has been passed to another utility command file.
3. An error has been encountered.

3A-17. Whenever a command file is active, it is the only source of keyboard input. This means that an INPUT statement in a user program, called by a command file that has additional lines, will take its input from the next line of the active command file. If the program terminates through an END, STOP, or EXIT statement, the next command file line is taken as keyboard input.

3A-18. A command file may transfer control to the first line of any utility command file, including itself, by commanding a return to the Console Monitor, followed by a line containing the name of the utility command file. The .CMD extension is not used. Subsequent command lines in the original command file will not be processed.

### 3A-19. The System Command File

3A-20. The system command file is activated only at system start-up time. After power-on, or pressing RESTART, the 1720A runs an internal self-test, loads FDOS, and searches the start-up system device for a file named COMMND.SYS. If it is found, the Console Monitor is then loaded with command file status active, and FDOS processes the command file as keyboard inputs to CONMON and any other mode selected as a result of the inputs. The system command file takes control only at start-up time. Control cannot be subsequently transferred to it. Figure 3A-1 illustrates this sequence.

### 3A-21. Utility Command Files

3A-22. A utility command file is a sequence of keyboard inputs stored in a file with a .CMD name extension. The effect is that the user can create simplified commands that operate directly from the Console Monitor.

3A-23. Control is transferred to a utility command file whenever the Console Monitor processes a file name which is found to have the .CMD extension. In addition, a command argument can be passed to a utility command file at the time it is called. It is generally good practice to include commands that return the system to the Console Monitor after completing the task.

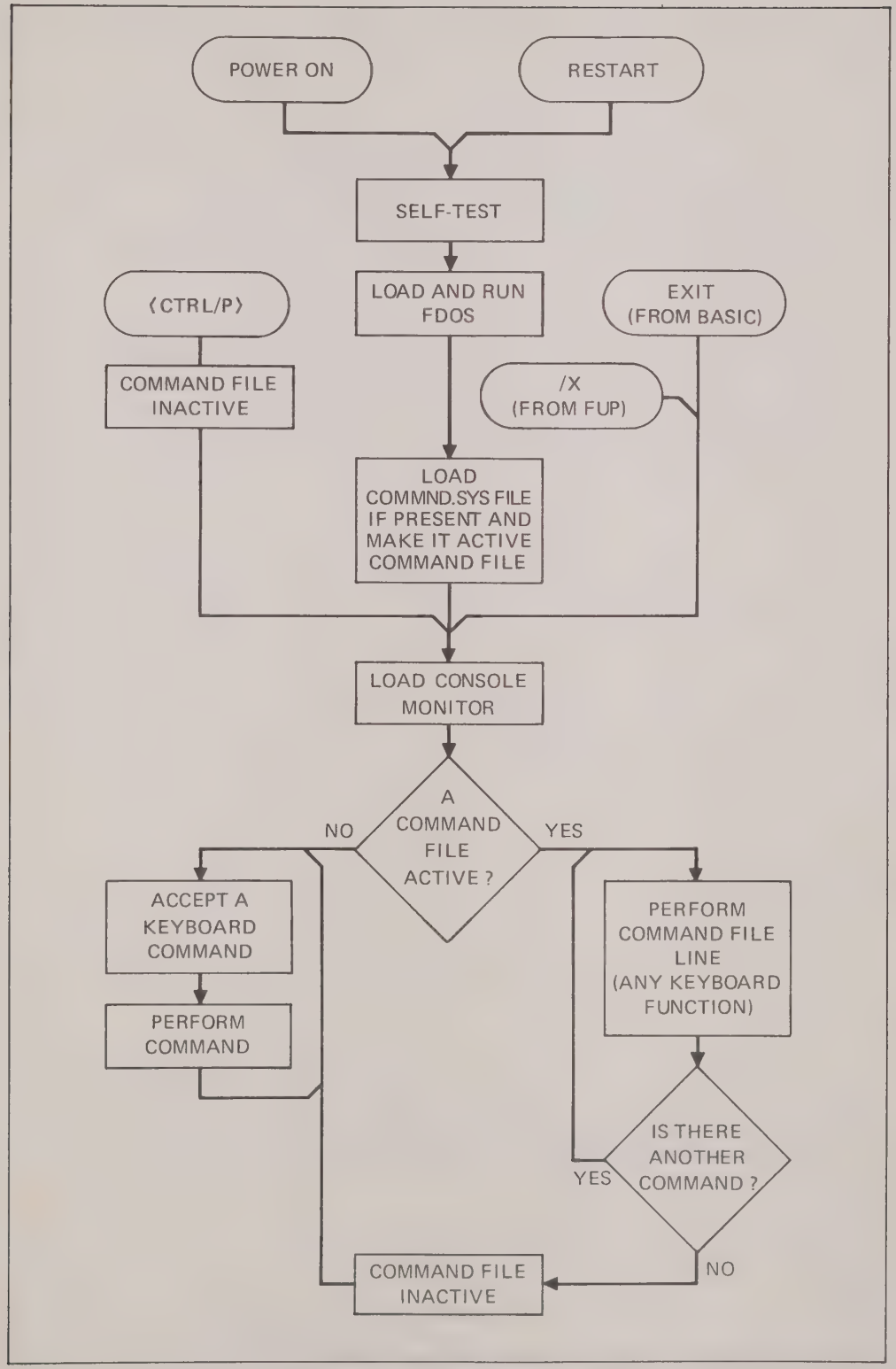



**Figure 3A-1. Start Sequence**

3A-24. Utility command files have all of the characteristics listed previously. The following points are true only for utility command files:

- The file name extension .CMD identifies a utility command file.
- Control is transfered to a utility command file from the Console Monitor by entering the name of the command file. The .CMD extension is not used.
- A utility command file may be called directly or from any active command file, including itself.
- An argument can be passed to a utility command file by including additional characters after the command file name when it is called. These characters must be separated from the file name by a space and must represent a valid keyboard input for the operation that will use them. Wherever a ? character appears in the command file, the argument is substituted. The ? character is ignored when an argument is not included. Examples of this follow.

### 3A-25. Command File Examples

3A-26. The following command file performs four tasks:

1. Sets some selected parameters on Serial Port 2 using the SET utility.
2. Verifies that the time clock has not experienced a power interrupt (or requests new time information using the TIME utility.
3. Loads the BASIC interpreter.
4. Loads the program TEST43, remaining in immediate mode.

```
SET
KB2: BR 1200 DB 7 PB EVEN EXIT
TIME
BASIC
OLD "TEST43"
```

3A-27. This would be a useful COMMND.SYS system command file during the development of program TEST43. Note that the words SET, EVEN, EXIT, TIME, and BASIC could have been abbreviated S, E, EX, T, and B respectively.

3A-28. If the 1720A used for program development contains an optional electronic disk, the COMMND.SYS file can be expanded to speed up operation during programming. In addition to the previous tasks, this command file copies some essential system software and the program TEST43 onto the electronic disk and assigns the disk as the system device. The programmer must then remember to save the revised TEST43, after further development, onto the floppy disk.

```
SET
KB2: BR 1200 DB 7 PB EVEN EXIT
TIME
FUP
ED0:/F
YES
ED0:=FDOS.SYS,CONMON.SYS,FUP.CIL,BASIC.CIL/B
ED0:=TEST43
```

```
EDO:/A
/X
BASIC
OLD "TEST43"
```

3A-29. After the program has been developed, the working disk that uses it might contain the COMMND.SYS system command file given below. This disk automatically begins running the program TEST43 whenever the 1720A is powered ON or RESTART is pressed. A Programmer Keyboard is not required unless the time clock needs to be reset.

```
TIME
BASIC
RUN "TEST43"
```

3A-30. The following command file loads the BASIC interpreter, runs program TEST1, and then runs program TEST2. If program TEST1 terminates through an EXIT statement, the command file needs an additional line to reload the BASIC interpreter before attempting to run TEST2.

```
BASIC
&<ESC>[2J<ESC>[1p<ESC>[4;6HStandby for Test 1.
RUN "TEST1"
&<ESC>[2J<ESC>[1p<ESC>[4;6HStandby for Test 2.
RUN "TEST2"
```

3A-31. The message lines in the previous example use display control codes to clear the screen, select double-size characters, position the display cursor, and display a message. This is useful for a large ASCII-format program to inform the operator that Test 1 or Test 2 is being loaded. Display codes are used directly, including the ESCape key, without the need for quotes or the CHR$ function as in a BASIC PRINT statement. ⟨ESC⟩ represents an ESCape key entry.

3A-32. The following command file uses FUP commands to display a directory of the system device whenever it is called and then return to the Console Monitor. If it is given the file name DIR.CMD, the system device directory can then be directly displayed from the Console Monitor by typing DIR.

```
FUP
/L
/X
```

3A-33. By adding a ? character to the DIR.CMD file, it gains the ability to accept an argument. In the following example, typing DIR ED$\phi$: will produce a directory listing of the electronic disk. Typing DIR will produce a directory of the system device, as in the previous example.

```
FUP
?/L
/X
```

### 3A-34. Creating Command Files

3A-35. Two ways are presented to create a command file. Using the File Utility Program (FUP), the file can be directly typed and stored. For longer command files, the

editing capabilities of BASIC are often useful. A BASIC program is presented that will generate and store a command file and a revised version of itself. Procedures for each of these methods follow:

3A-36. For a short command file, use FUP to directly type in the file.

1. After a CONMON prompt #, type F to select the File Utility Program.

2. Type MFØ: (File name) =KBØ:. For the file name, use COMMND.SYS for a system command file, or (command name).CMD for a utility command file.

3. Type each command input to be stored on a separate line. Use [DELETE] to correct errors.

4. Use [ESC] directly to initiate display control sequences.

5. Use [CTRL] with the letters given in Table 3A-2 to include some of the characters and control codes below character number 27 in a display string.

*NOTE*

*If the command file requires codes below character number 27 that are not included in Table 3A-2, a BASIC program must be used to generate the file.*

6. Use two [RETURN] entries after the last line, when the last line is a display string introduced by &.

7. Terminate the file with ⟨CTRL/Z.⟩

8. Type /X to return to the Console Monitor.

9. To use the command file, press RESTART on the front panel for a system command file, or type the assigned command name for a utility command file. (Do not include the .CMD extension.)

**Table 3A-2. Control Codes Available for FUP Generated Command Files**

| CONTROL CODE | CHARACTER OR FUNCTION | CONTROL CODE | CHARACTER OR FUNCTION |
|---|---|---|---|
| A | $\beta$ | K | line feed |
| B | $\gamma$ | L | line feed |
| D | $\epsilon$ | N | $\phi$ |
| E | $\zeta$ | O | $\pi$ |
| F | $\eta$ | R | $\tau$ |
| G | beep | V | $\psi$ |
| H | backspace | W | $\omega$ |
| I | horizontal tab | Y | $\Omega$ |

3A-37. For a longer command file, use the BASIC program given in Figure 3A-2. Refer to the 1720A Fluke BASIC Programming Manual for information on creating and storing BASIC programs. This program creates two files: the command file and a copy of this program with the name (command name).BAL. Note the following points:

- Use one DATA statement for each command file line.
- In place of (file name) in line 10010 use COMMND.SYS for a system command file, or (command name) .CMD for a utility command file.

- In place of (file name) in lines 10120 and 10130, use `COMMND.BAL` for a system command file, or (command name).`BAL` for a utility command file.
- If the last command file line is a display string introduced by &, add one more DATA statement with a null string ( `DATA ""` ).
- Because DATA statements cannot contain control codes such as ESCape, use a ^character to indicate CTRL, and a $ character to indicate ESCape.

```
1000     ON ERROR GOTO 10130
1010     DATA    "    "
1020     DATA    "    "
1030     DATA    "    "
10000    DATA "END"
10010    OPEN "(file name)" AS NEW FILE 1
10020    READ A$ \ IF A$= "END" GOTO 10110
10030    C$ = ""
10040    FOR I%=1% TO LEN (A$)
10050      B$=MID (A$,I%,1%)
10060      IF B$= "^" THEN I%=I%+1%\B$=CHR$(ASCII(MID(A$,I%,1%))-64%
10070      IF B$= "$" THEN B$ = CHR$(27)
10080      C$ = C$+B$
10090    NEXT I%
10100    PRINT #1, C$ \ GOTO 10020
10110    CLOSE 1
10120    KILL "(file name)" \ SAVEL "(file name)" \ GOTO 10140
10130    IF ERL = 10120 AND ERR = 305 THEN SAVEL "(file name)" ELSE OFF ERROR
10140    END
```

**Figure 3A-2. Command File Generator Program**

## 3A-38. CONTROL CODES

3A-39. Control character codes are formed by holding the CTRL key depressed while typing another letter, just as when using the SHIFT key. Some of these codes are intercepted by FDOS when detected and used to initiate specific actions. Table 3A-3 summarizes these codes and the resulting actions.

**Table 3A-3. Control Code Summary**

| CODE | RESULT |
|---|---|
| ⟨CTRL/C⟩ | Stop program |
| ⟨CTRL/P⟩ | Abort program and return to CONMON |
| ⟨CTRL/Q⟩ | Continue display after stop |
| ⟨CTRL/S⟩ | Stop display |
| ⟨CTRL/T⟩ | Clear screen, reset display modes, cursor to upper left |
| ⟨CTRL/U⟩ | Delete current line |
| ⟨CTRL/Z⟩ | End-of-file code for ASCII files |

# Section 3B
# CONMON

## 3B-1. INTRODUCTION

3B-2. The Console Monitor (CONMON) is a machine-language system software file that is supplied on the System Disk with the file name CONMON.SYS. When loaded into Main Memory it processes keyboard inputs, giving the user access to utility programs and programming languages. Each utility program and language includes a command to return to CONMON.

## 3B-3. ACCESS TO CONMON

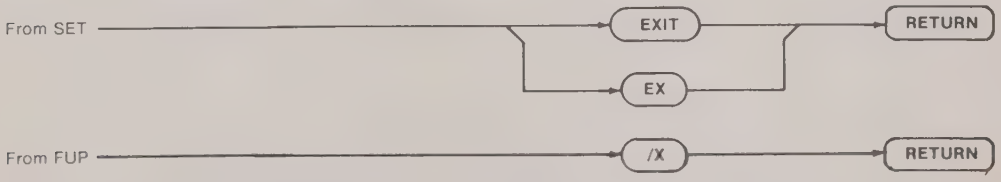
3B-4. Several different paths are provided for gaining access to CONMON. The start-up sequence terminates with the loading of CONMON when a system command file is not provided. Each utility program and language includes a command to return to CONMON. Also, a direct access method is provided that overrides all other operations. Each of these is discussed below.

### 3B-5. Start-Up Access

3B-6. Refer to Figure 3A-1. Whenever power is turned on, or RESTART is pressed, the 1720A loads CONMON after completing a self-test, loading FDOS, and checking for a system command file. If the file COMMND.SYS is not found on the system device, the operation stops with CONMON loaded and command file status inactive.

3B-7. To save a system command file without allowing it to intercept the start-up sequence, use FUP to rename it to COMMND.CMD. It will then be ignored at start-up time. However, it can be used any time COMMND is given as an input to CONMON.

### 3B-8. Access From Utility Programs




3B-9. Each utility program includes a command to return to CONMON. When the command is entered, the utility program properly completes and terminates any operation that was in progress before requesting FDOS to load CONMON. This process deletes the utility program from Main Memory.

### 3B-10. Access From BASIC

3B-11. When the BASIC interpreter processes an EXIT statement, all open files are closed and terminated and a request is made to FDOS to load CONMON. This process deletes BASIC and the user program that was in memory. The EXIT statement may be part of a running program, or may be directly executed as an immediate mode command.

### 3B-12. Direct Access

3B-13. Whenever a ⟨CTRL/P⟩ entry from the keyboard is detected by FDOS, any other operation in progress is immediately suspended, the file CONMON.SYS is loaded from the System Device, and control is transferred to it. Other Main Memory contents (except FDOS) are no longer usable. If a file was being written to the floppy or electronic disk, it will appear on the file directory, but may not be accessible to FDOS if the ⟨CTRL/P⟩ entry interrupted file completion and termination. This direct access method should not generally be used except as a back-up when other methods do not work and when the user is certain that the operation in process is not writing to a file.

### 3B-14. USING CONMON

3B-15. CONMON displays the message:

```
Console Monitor Version 1.1
#
```

*NOTE*

*Verify that the version number is 1.1. If it is not, check the front of this manual for supplemental information, or call a Fluke Customer Service Center for advice.*

3B-16. CONMON processes inputs in the following way:

1. A single letter entry of T, S, F, or B is converted internally to TIME, SET, FUP, or BASIC respectively.

2. The directory of the system device is searched for a file with the entered name, having a .CMD extension. If found, it is loaded into memory, command file status in FDOS is set active, and the contents of the file are processed as if they were received as inputs from the keyboard.

3. If the .CMD file is not found, the directory of the system device is searched for a file of the entered name, having a .CIL extension. If found, it is loaded into memory as a machine language program and control is transferred directly to it.

4. If neither the .CMD file nor the .CIL file is found, CONMON displays:

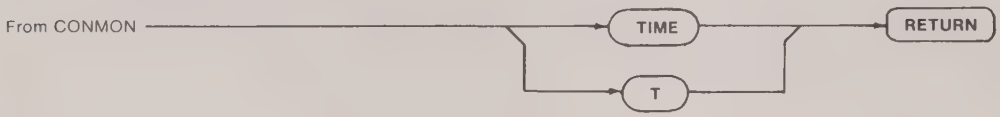
```
?File not found
```

# Section 3C
# TIME

## 3C-1. INTRODUCTION

3C-2. The Time and Date Set Utility (TIME) is a machine-language utility software file supplied on the System Disk with the file name TIME.CIL. When loaded into Main Memory, it requests information from the user to set the time and date of the internal time clock. When called from an active command file, it does not request input unless the clock status shows that the clock has lost power since the last setting by TIME.

## 3C-3. ACCESS TO TIME




3C-4. TIME is accessable through CONMON by typing TIME or T.

## 3C-5. EXITING TIME

3C-6. TIME may be exited by entering ⟨CTRL/P⟩. When the requested information is entered, control automatically returns to CONMON without the need for a separate command entry.

## 3C-7. USING TIME

3C-8. TIME displays the message:

```
TIME (time)(date)
Time Version 1.0
Enter date:  DD-MM-YY
```

*NOTE*

*Verify that the version number is 1.0. If it is not, check the front of this manual for supplement information, or call a Fluke Customer Service Center for advice.*

3C-9. Type in the date, numeric month, and the year (two digits). The entries may be separated by any non-numeric character. Complete the entry with RETURN. TIME responds with:

```
Enter time:  HH-MM
```

3C-10. Type in the time, in 24-hour form, starting with the hour, followed by the minutes. The entries may be separated by any non-numeric character. Complete the entry with RETURN, and TIME returns control to CONMON.

3C-11. The 24-hour day starts at 00:00 (midnight) and ends at 23:59 (just before midnight). If the time is at or beyond midnight and before 1:00 A.M., subtract 12 hours. If the time is at or beyond 1:00 P.M., and before midnight, add 12 hours. Otherwise, use the time unchanged. Three examples:

| TIME | 24-HOUR TIME |
|---|---|
| 12:30 A.M. | 00:30 |
| 9:45 A.M. | 09:45 |
| 4:20 P.M. | 16:20 |

3C-12. The clock does not compensate for leap years. Whenever February 29 occurs, it must be manually entered. It if occurred on a day the 1720A was not being used, the date will be ahead by one until it is corrected.

3C-13. The clock maintains a status flag that is set whenever power is first applied to it. The clock is supported by the internal battery even when the electronic disk is not. The TIME utility resets this flag after setting time and date information into the clock. Whenever TIME is called from an active command file, this status flag is checked. If it is found to be reset, no input is requested. This feature allows the use of a system command file that checks the clock for interruption without requiring keyboard input, except when the clock information is known to be wrong.

# Section 3D
# SET

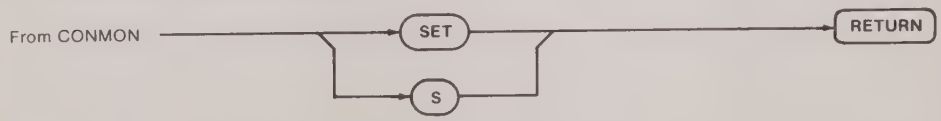
## 3D-1. INTRODUCTION

3D-2. The Set Serial Port Utility (SET) is a machine-language utility software file supplied on the System Disk with the file name SET.CIL. When loaded into Main Memory it allows the user to reconfigure the parameters used by FDOS in communicating through each of the two RS-232-C serial ports.

*NOTE*

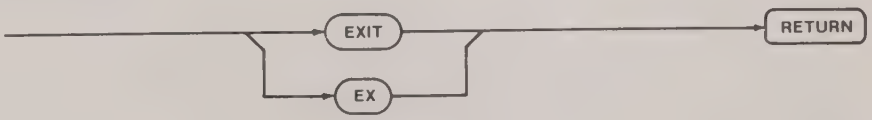
*The software supporting the RS-232 serial ports is not intended to handle binary data. Line and file terminators, as well as other control codes, are tested for and used to identify specific operations.*

## 3D-3. ACCESS TO SET




3D-4. SET is accessible through CONMON by typing `SET` or `S`.

## 3D-5. EXITING SET




3D-6. SET may be exited at any time by entering `EXIT` or `EX` in response to the SET prompt *.

## 3D-7. USING SET

3D-8. SET displays the message:

```
SET (time)(date)
Set Version 2.0
*
```

*NOTE*

*Verify that the version number is 2.0. If it is not, check the front of this manual for supplemental information, or call a Fluke Customer Service Center for advice.*

3D-9. After first specifying a port, SET allows the user to set any of the following characteristics for the port:

Baud Rate
Number of Data Bits
Parity
Number of Stop Bits
End of Line Character
End of File Character

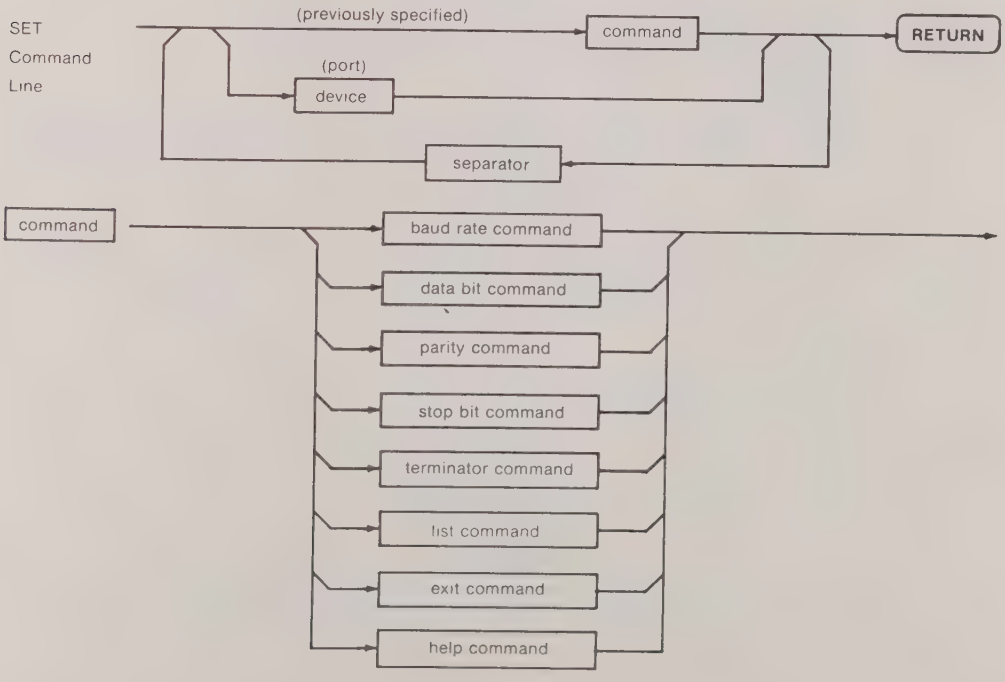
3D-10. In addition, SET displays the current configuration of the selected port or a summary of SET commands.

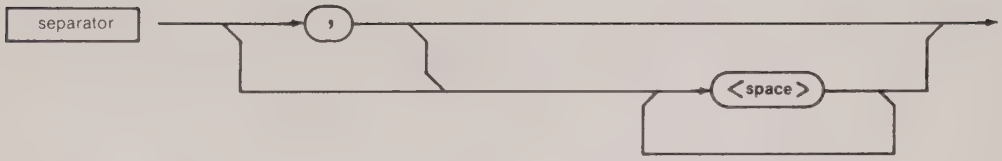
### 3D-11. SET Command Structure

3D-12. The SET utility program uses a flexible and straightforward command structure. After selecting a device, the user may list the current settings and change selected ones in any order.

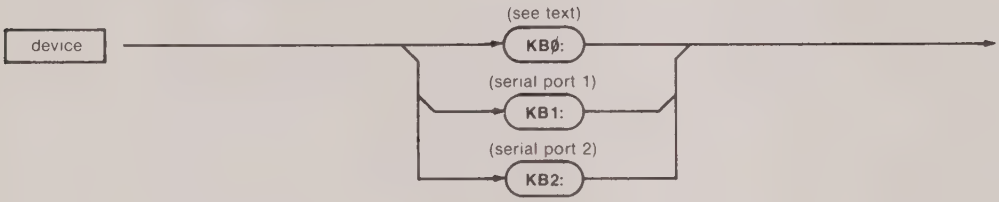
- Commands may be entered singly or may be combined into a multiple command line.
- Both upper and lower case entries are accepted.
- Commands become effective when the command line is terminated by RETURN.
- All SET commands may be automated through command files.
- Parameters controlled by SET may be independently set for each serial port.

3D-13. The syntax diagrams given below define the structure of SET commands:






### 3D-14. Device Specification




3D-15. SET requires that a port device be selected prior to accepting other commands.

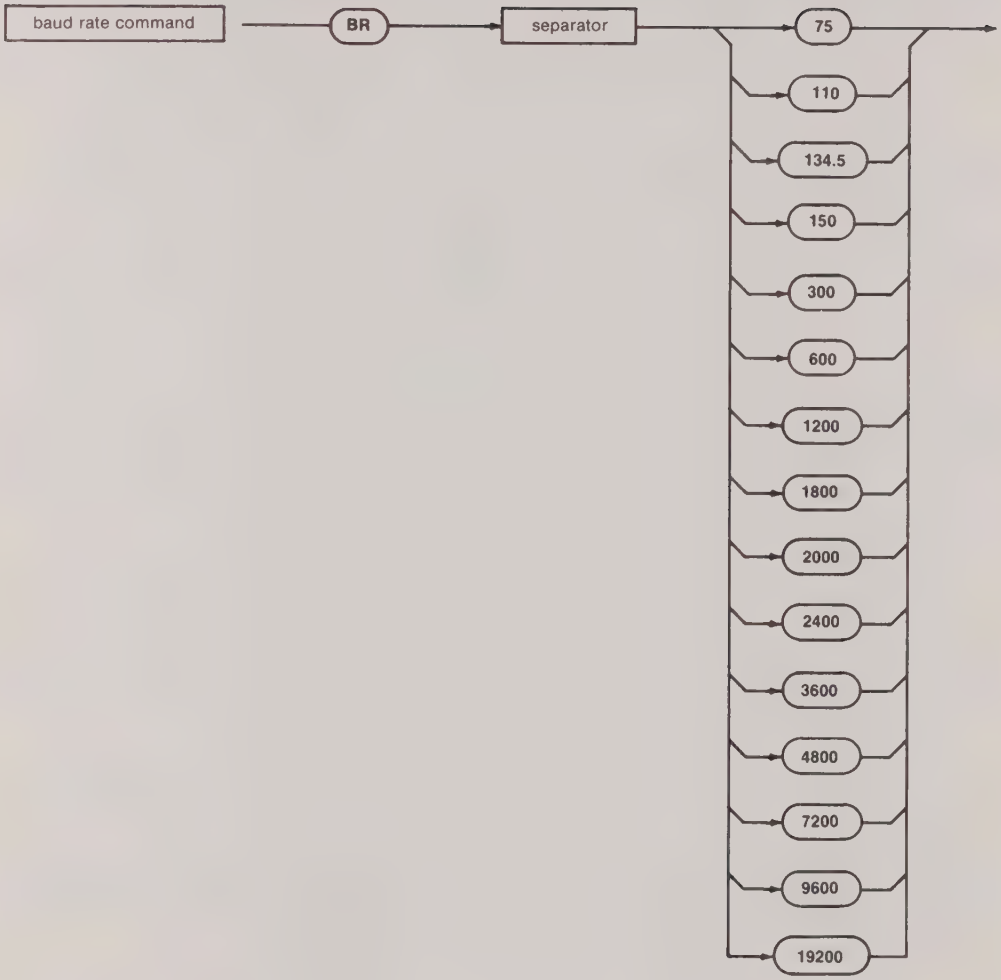
- Once selected, subsequent commands affect the specified port until a different port is selected.
- `?Unknown command` is displayed when a device is selected other than KBØ:, KB1:, or KB2:.
- `?Illegal device` is displayed when KB2: is selected while an external terminal is in use. See the discussion of KBØ: below.
- `?No device specified` is displayed when a command is entered before a port is selected.

3D-16. KBØ: identifies the console device. All command inputs except baud rate are ignored for KBØ:. Normally, the console device is the Programmer Keyboard for input, and the display for output.

*NOTE*

*The baud rate of the 1720A Programmer Keyboard and display is 4800. Setting KBØ: to any other baud rate immediately disables both the programmer keyboard and the display. This condition is remedied by pressing RESTART.*

### 3D-17. Setting Baud Rates



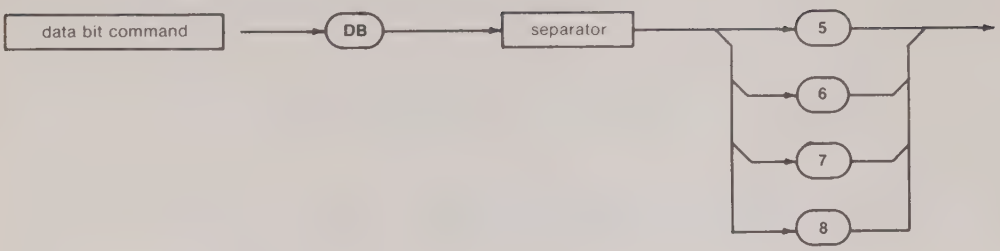

3D-18. The baud rate command changes the baud rate of the selected port to that specified by the command argument. Baud rate is the bit transfer rate in bits per second, including start, stop, parity, and unused bits.

- The baud rate change takes effect when RETURN is pressed.
- `?Bad argument` is displayed if a baud rate is entered that is not one of the 15 values given above.

3D-19. The following example sets 2400 baud for Serial Port 2, and 300 baud for Serial Port 1:

```
*KB2: BR 2400, KB1: BR 300 RETURN
```

### 3D-20. Setting Character Length



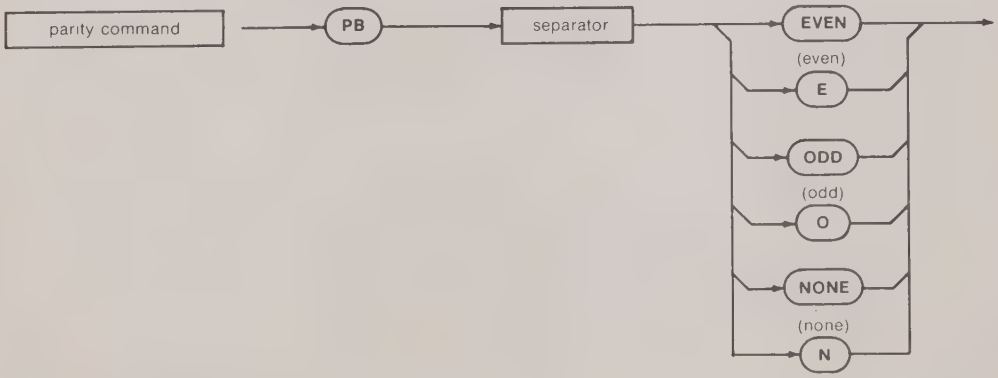

3D-21. The data bit command sets the number of data bits that will be included in each character.

- When character length is shorter than the actual data, the lower (least significant) data bits are used.
- When character length is longer than the actual data, the remaining (most significant) bits are set to zero.
- `?Bad argument` is displayed if a word length is entered that is not one of the four values given above.

3D-22. The following example selects Serial Port 2, sets its baud rate to 1200, and sets the character length to 7 bits:

```
*KB2: BR 1200, DB 7 RETURN
```

### 3D-23. Defining a Parity Bit




3D-24. The parity command defines a parity bit to be generated and checked for the selected port. Parity is an error detection scheme that uses an extra bit appended after the last data bit of each word. This bit is set so that the total number of 1-bits in each word is always even or odd.

- The command argument `NONE` or `N` eliminates parity generation and checking.
- `?Bad argument` is displayed if a command argument is entered that is not one of the six given above.

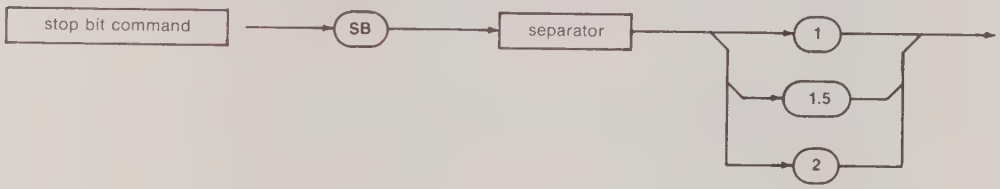
3D-25. During input, parity is checked as defined by SET for each character. If an error is detected, it is identified to FDOS as a device error. In a BASIC program, this returns error number 311. This error can only be cleared by closing the channel associated with the serial port.

3D-26. During output, parity is generated as defined by SET, and appended to each character. Parity is always in addition to the data bits.

3D-27. The following example selects Serial Port 2, sets its baud rate to 1200, sets the word length to 7 data bits, and defines even parity:

```
*KB2: BR 1200 DB 7, PB EVEN RETURN
```

### 3D-28. Defining Character Spacing



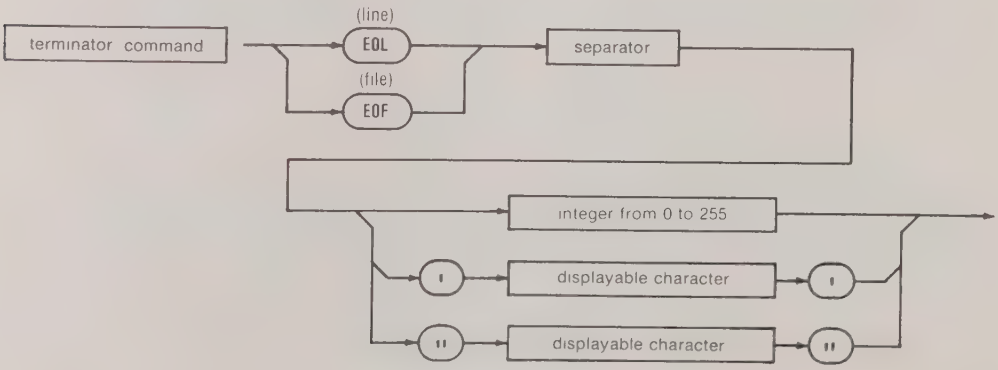

3D-29. The stop bit command defines the number of bit-cell time periods to be allocated as a minimum spacing between characters transmitted to external equipment that requires additional settling or synchronization time.

- This command does not affect incoming data.
- `?Bad argument` is displayed if an argument is entered that is not one of the three given in the syntax diagram.

3D-30. The following example defines a minimum transmission word spacing of 1.5 bit-cell time periods for Serial Port 1:

```
*KB1: SB 1.5 RETURN
```

### 3D-31. Defining Terminator Characters




3D-32. The terminator command defines a character that will be used to identify the end of a line or file. When received with incoming data, a terminator will generate an interrupt if enabled by the user program.

- `?Argument out of range` is displayed if an integer argument greater than 255 is entered.
- `?Bad argument` is displayed if an argument is entered that otherwise does not meet the requirements of the syntax diagram.
- Line and file terminator characters are handled as described in the following paragraph.

3D-33. The end-of-line character defined by SET only affects data input:

- All Carriage Return and Line Feed characters are deleted.
- A Carriage Return, Line Feed sequence is then appended after each occurrence of the line terminator character.
- If Line Feed or Carriage Return is the terminator, this process does not duplicate it.
- Resulting data is compatible with 1720A internal format.
- During output, data is not changed by the line terminator defined by SET.

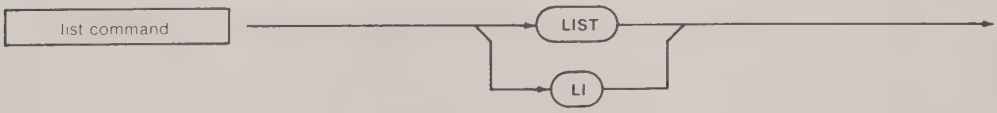
3D-34. The end-of-file character defined by SET affects both input and output:

- During input, each occurrence of the file terminator defined by SET is deleted, and ⟨CTRL/Z⟩ (character 26) is put in its place.
- During output, each occurrence of a ⟨CTRL/Z⟩ character is deleted, and the file terminator defined by SET is put in its place.

3D-35. The following example defines a question mark as line terminator, and the ASCII EOT code (⟨CTRL/D⟩) as a file terminator for the port that was previously addressed:

```
eol '?' eof 4 RETURN
```

### 3D-36. Listing Port Configuration




3D-37. The list command displays the configuration of the currently selected port.

- Displayable terminator characters are shown as their decimal value and as the character.

3D-38. The following example lists the configuration of Serial Port 2:

```
*KB2: LIST RETURN
```

3D-39. The resulting display from this command might appear as follows:

```
Device          KB1:
Baud Rate       2400
Data Bits       7
Parity          even
Stop Bits       1
End of Line     10
End of File     26
```

### 3D-40. The SET Help Display

3D-41. The help command clears the screen and displays a summary of all SET commands with examples.

### 3D-42. SET ERROR MESSAGES

4D-43. SET provides error messages that aid the user in recognizing faulty command inputs. Table 3D-1 summarizes and defines these errors.

**Table 3D-1. SET Error Messages**

| MESSAGE | MEANING |
|---|---|
| ?Argument missing | A command was entered without the argument necessary to complete its meaning. |
| ?Argument out of range | A command argument was entered which was beyond the range of acceptable values. |
| ?Bad argument | A command argument was entered which was not in the list of acceptable arguments for that command. |
| ?Illegal device | KB2: was selected when an external terminal was in use. (See Section 4A.) |
| ?No device specified | A command was entered before specifying a device. |
| ?Unknown command | A command was entered that was not recognized. |

### 3D-44. SET COMMAND SUMMARY

3D-45. Table 3D-2 summarizes SET commands and command arguments.

**Table 3D-2. SET Commands and Arguments**

| FUNCTION | COMMAND | ARGUMENTS |
|---|---|---|
| Select Port Device | KB0:<br>KB1:<br>KB2: | |
| Define Baud Rate | BR | 75 600 3600<br>110 1200 4800<br>134.5 1800 7200<br>150 2000 9600<br>300 2400 19200 |
| Define Data Bits Per Character | DB | 5<br>6<br>7<br>8 |
| Define Parity Bit | PB | EVEN<br>E<br>ODD<br>O<br>NONE<br>N |
| Define Stop Bits For Character Spacing | SB | 1<br>1.5<br>2 |
| Define End of Line Terminator | EOL | Integer 0 through 255<br>' character '<br>" character " |
| Define End of File Terminator | EOF | Integer 0 through 255<br>' character '<br>" character " |
| Return to CONMON | EXIT<br>EX<br><CTRL/P> | |
| List Port Configuration | LIST<br>LI | |
| Ignore this Command Line | <CTRL/C> | |
| Display Command Summary | ? | |

# Section 3E
# FUP

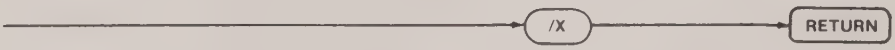
## 3E-1. INTRODUCTION

3E-2. The File Utility Program (FUP) is a machine-language utility software file supplied on the System Disk with the file name FUP.CIL. When loaded into main memory it gives the user control over files on floppy disks and on the optional electronic disk. The flexible structure of FUP also provides other useful capabilities. The examples provided in this section illustrate only some of the ways this multi-purpose tool can be used.

## 3E-3. ACCESS TO FUP

3E-4. FUP is accessible through CONMON by typing FUP or F.

## 3E-5. EXITING FUP




3E-6. FUP may be exited at any time by entering **/X** in response to the FUP prompt *.

## 3E-7. USING FUP

3E-8. FUP displays the message:

**FUP** (time)(date)
**File Utility Program Version 1.1**
*

*NOTE*

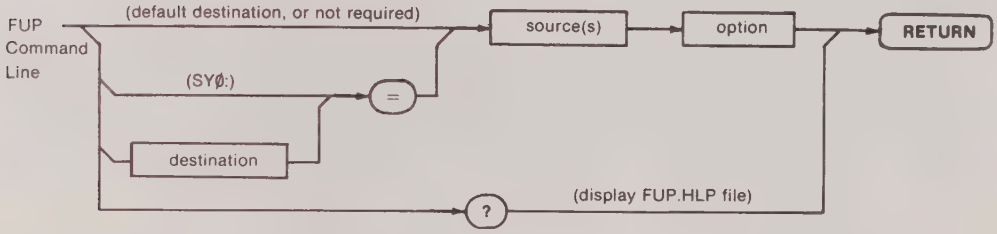
*Verify that the version number is 1.1. If it is not, check the front of this manual for supplemental information, or call a Fluke Customer Service Center for advice.*

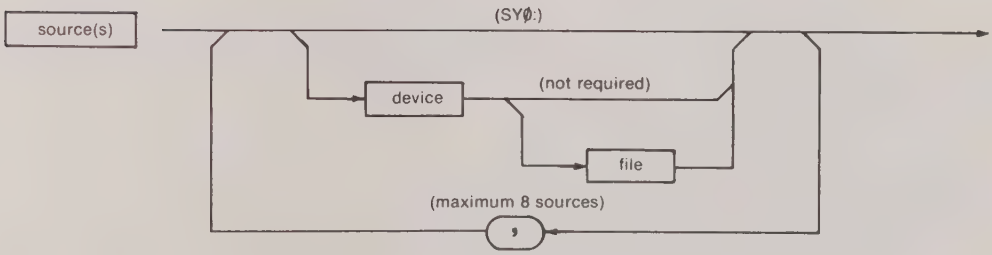
## 3E-9. FUP Command Structure

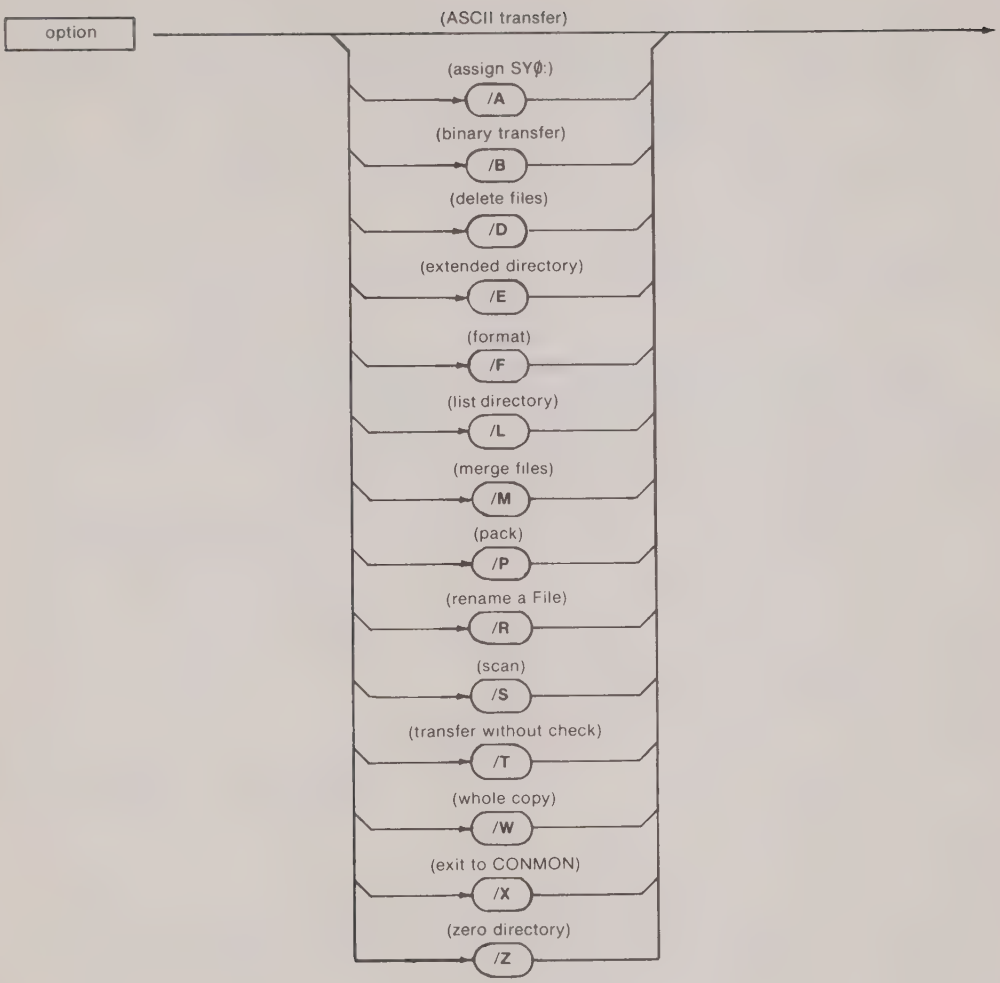
3E-10. A FUP command sets up a communication channel between a specified destination device or file, and one or more source devices or files. A variety of command options are available for specific additional actions.

- Each command line must contain only one FUP command.
- Both upper and lower case entries are accepted.
- Commands become effective when the command line is terminated by RETURN.
- The pseudo-device MMØ: (main memory) can be used in FUP as temporary file storage for copying without the optional electronic disk.
- All FUP commands may be automated through command files.

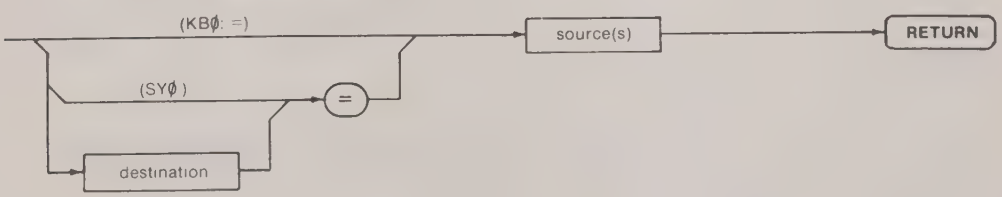
3E-11. The syntax diagrams given below define the structure of FUP commands:








### 3E-12. ASCII Transfer (no option)




3E-13. A FUP command without a specified option sets up an ASCII communication channel between the specified destination and source(s).

- Up to eight sources may be specified.
- Transfer of a ⟨CTRL/Z⟩, character number 26, identifies the end of each source. Transfer then proceeds with the next source on the list, or terminates.
- When the destination is a named file, each source overlays the previous one. The resulting destination file contains only the last specified source. To merge multiple files, refer to the discussion of the FUP /M option.

- When the destination is a file device (`MF0:`, `ED0:`, or `MM0:`) and no destination file name is specified, the names of the source files are used. If any source is not a file device (no name can be identified), the resulting destination file will have no name. (See the FDOS discussion of the no-name file.)

- When the source is a file device and a source file is not specified, the source device is searched for a no-name file.

- Once a system device is specified in the source list, that device is assumed until a different device is specified.

**CAUTION**

**If an existing file is specified as destination, it will be deleted without warning and replaced by information from the specified source.**

3E-14. The examples given below show the variety of ways in which this FUP command construct is useful. A short description precedes each example.

The display (`KB0:`) is specified as destination, and `T44.BAS` on the floppy disk (`MF0:`) as source. This will display the file. Use Page Mode if the file is longer than 16 lines:

`KB0:=MF0:T44.BAS` RETURN

This is equivalent to the previous one if the floppy disk is the System Device. The use of defaults eliminates 13 of the 17 keystrokes:

`T44` RETURN

This copies `PROGRM.38` from the floppy disk into temporary storage in main memory using the same file name:

`MM0:=MF0:PROGRM.38` RETURN

This completes the copy operation started above, after exchanging floppy disks. `PROGRM.38` is copied onto the floppy disk (`MF0:`) from temporary storage in main memory (`MM0:`) using the same file name:

`MF0:=MM0:PROGRM.38` RETURN

This produces a duplicate copy of `TEST.C41` on the System Device under the name `TEMP.BAS.`:

`TEMP=TEST.C41` RETURN

A file named `MEMO.23` is created on the floppy disk (`MF0:`) from keyboard (`KB0:`) inputs. A ⟨CTRL/Z⟩ input is used to terminate the file:

`MF0:MEM0.23=KB0:` RETURN

This example transfers ASCII data to the device, such as a printer, connected to RS-232 Serial Port 2 (`KB2:`) from keyboard (`KB0:`) inputs. A ⟨CTRL/Z⟩ input terminates this mode:

`KB2:=KB0:` RETURN

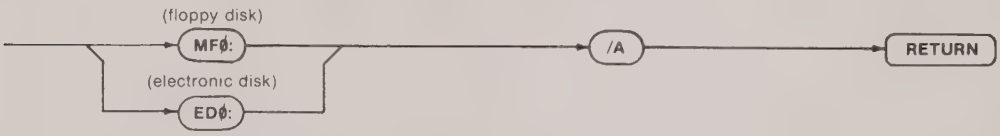
This example sends to the device connected to Serial Port 1 a copy of `TEST.M6` on the System Device:

`KB1:=TEST.M6` RETURN

This sends to the device connected to serial port 2, copies of `TEST.Q4` on the System Device, and `TEST.Q3` and `TEST.Q2` from the floppy disk (`MF0:`). Note that `MF0:` is not specified for `TEST.Q2` since it was previously specified.

```
KB2:=TEST.Q4,MF0:TEST.Q3,TEST.Q2
```
RETURN

### 3E-15. Assign the System Device (/A)




3E-16. The /A FUP command option is used to assign the default file device, called the System Device. Refer to the discussion of FDOS for information on the System Device.

3E-17. The following example assigns the optional electronic disk as System Device.

```
ED0:/A
```
RETURN

### 3E-18. Binary Transfer (/B)

3E-19. The /B FUP command option is used to transfer data that contains binary code. This includes system and utility software files, lexical form BASIC programs (.BAL extension), and virtual array files.

- Up to eight source files may be specified.
- When the destination is a named file, the resulting file will contain only the last specified source file. Binary files cannot be merged.
- When the destination is a file device (`MF0:`, `ED0:`, or `MM0:`) and no destination file name is specified, the names of the source files are used.
- Once a system device is specified in the source list, that device is assumed until a different device is specified.

**CAUTION**

**If an existing file is specified as destination, it will be deleted without warning and replaced by the source file.**

3E-20. This example transfers onto the floppy disk (`MF0:`) the virtual array file `T4.DAT` from the electronic disk (`ED0:`). A date-coded name is assigned to the file copy by this transfer.

```
MF0:220981.T4D=ED0:T4.DAT/B
```
RETURN

3E-21. This transfers some system and utility software files into temporary storage in main memory.

```
MM0:=FDOS.SYS,CONMON.SYS,SET.CIL,FUP.CIL/B
```
RETURN

3E-22. This completes the copy operation started above after exchanging floppy disks. The files in temporary storage in main memory ( **MMØ:** ) are copied onto the floppy disk ( **MFØ:**) using the same file names.

```
MFØ:=MMØ:FDOS.SYS,CONMON.SYS,SET.CIL,FUP.CIL/B  RETURN
```

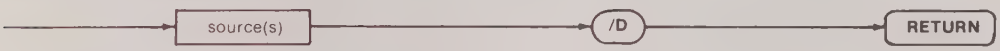
*NOTE*

*A new disk must be formatted (/F option) before copying onto it.*

*NOTE*

*Be sure that a new System Disk created through FUP has compatible system and utility software files. Compatible version numbers are identified in the front of published Fluke manuals and manual addendums. Failure to observe this will produce unpredictable results for which Flue cannot provide analytical, diagnostic, or other software support, except for identification of compatible combinations.*

### 3E-23. Deleting Files (/D)




3E-24. The /D FUP command option is used to delete specified files.

- Up to eight files may be deleted at once.
- A single command may delete files from more than one file device.
- Deleting a file leaves a gap in the file structure. Refer to the Pack (/P) command option.
- When no file is specified, the no-name file is deleted.

3E-25. This example deletes the file **TEST6.BAS** from the System Device.

```
TEST6.BAS/D  RETURN
```

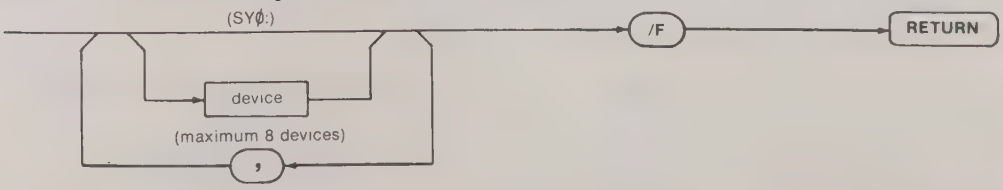
3E-26. This deletes the BASIC Interpreter system software file from temporary storage in main memory to make room for other files.

```
MMØ:BASIC.CIL/D  RETURN
```

3E-27. The following example deletes the user program **TEST.16** and the virtual array file T16.DAT from the floppy disk ( **MFØ:**). Note that **MFØ:** is only specified once.

```
MFØ:TEST.16,T16.DAT/D  RETURN
```

### 3E-28. Format, Verify, and Zero File Device(s) (/F)




3E-29. The /F FUP command option is used to prepare a floppy or electronic disk to receive files. Formatting is a process of writing identifying codes and standard data

patterns throughout the disk in a defined format. During this process, each data block is verified for data integrity. A message is displayed if a block requires one or more retries, or is unusable. A file directory is then set up in proper format with all entries set to zero.

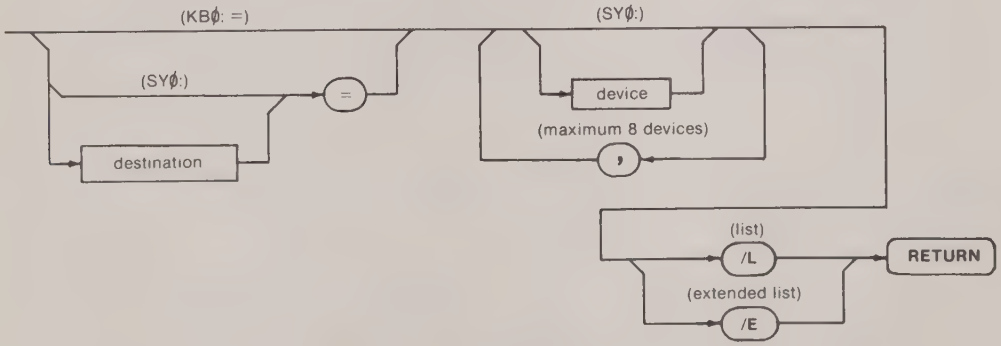
- Because formatting deletes all files from the specified device, FUP requests an affirmative before proceeding. Entries accepted as affirmative are **YES**, **Y**, **yes**, and **y**.
- The affirmative response must be on a separate line. It is required even when **/F** is used in a command file.
- Formatting a floppy disk (**MFØ:**) also causes the pseudo-device main memory (**MMØ:**) to be zeroed, deleting any files that were in temporary storage.
- The electronic disk must be formatted before use.
- The main memory temporary storage (**MMØ:**) does not require formatting before use.

3E-30. This example formats, verifies, and zeros a floppy disk:

**MFØ:/F** RETURN

3E-31. The message **Really format MFØ?** will then be displayed. An entry of **YES**, **Y**, **yes**, or **y** will allow formatting to proceed.

### 3E-32. Listing a Directory (/L or /E)




3E-33. The /L and /E FUP command options are used to list a directory of one or more file storage devices.

- The listing is sent to the display when no other destination is specified.
- The extended list includes all empty file areas remaining after deleting files. See the **/P** FUP command option for a technique of consolidating these empty areas.

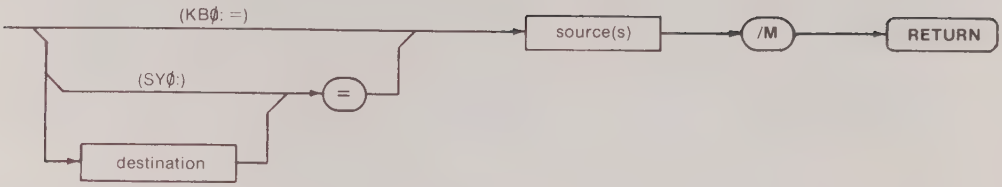
3E-34. The following example displays an extended directory list of the System Device:

**/E** RETURN

3E-35. The following example sends to Serial Port 2 (**KB2:**) a directory list of the floppy disk (**MFØ:**) and the electronic disk (**EDØ:**):

**KB2:=MMØ:,EDØ:/L** RETURN

### 3E-36. Merging ASCII Files (/M)




3E-37. The /M FUP command option is used to merge up to eight ASCII source files into one destination file.

- The source files remain intact unless the destination file uses one of their names.
- Once a file device is specified in the source list, that device is assumed until a different device is specified.
- When the destination file is not given a name, a no-name file is created.
- When the destination is a non-file device (such as `KB2:`), the /M option has the effect of removing the ⟨CTRL/Z⟩ character from the end of all but the last file.
- System and utility software files, lexical form BASIC programs (.BAL extension), and virtual array files cannot be merged.

**CAUTION**

**Make sure that the line numbers in BASIC programs are different before merging them together. The BASIC Interpreter will allow only one occurance of each line number. When a duplicate line number is encountered by BASIC during a program load into main memory, the previous line of that number is deleted.**

3E-38. The following example creates a file on the floppy disk (`MFØ:`) called `PROGRM.T44`. The new file will be composed in sequence of the contents of `TEST1.BAS` and `TEST2.BAS` on the floppy disk, and `TEST4.BAS` on the electronic disk. The original files all remain intact.

```
MFØ:PROGRM.T44=MFØ:TEST1,TEST2,EDØ:TEST4/M  RETURN
```

3E-39. The following example merges two files and deletes one of them by reusing the same file name. The new file `TEST4.BAS` is composed of the old file `TEST4.BAS` and the file `TEST1.BAS`. The file `TEST1.BAS` remains intact.

```
TEST4=TEST4,TEST1/M  RETURN
```

3E-40. The following technique allows the user to directly append keyboard input to the end of an existing file without creating a new file. The destination file `DATA$C.832` consists of the old file `DATA$C.832` followed by keyboard (`KBØ:`) inputs. A keyboard entry of ⟨CTRL/Z⟩ terminates the keyboard portion of the input. The old file is deleted by the creation of the new file with the same name.

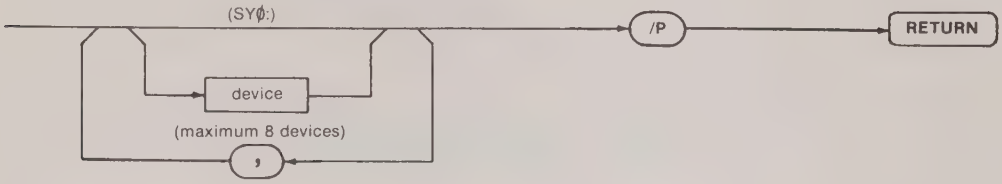
```
DATA$C.832=DATA$C.832,KBØ:/M  RETURN
```

3E-41. The following example is similar to the previous one, except that the keyboard input is placed at the beginning of the file. An entry of ⟨CTRL/Z⟩ terminates the

keyboard (KBØ:) portion, and copies the file **DATA.T26** after it. The old file is replaced by the new file with the same name.

**DATA.T26=KBØ:,DATA.T26/M** RETURN

3E-42. Packing a File Device (/P)




3E-43. The /P FUP command option is used to restructure a file device so that the unused areas left by file deletions are collected into a contiguous area at the end of the file structure.

- Since files are contiguous, packing often will make room for a file that previously would not fit.
- The extended list (/E option) will show whether packing will be of benefit.
- Packing has no effect on the integrity or size of files.

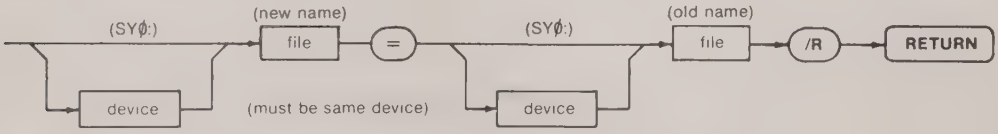
3E-44. The following example will pack the System Device:

**/P** RETURN

3E-45. The following example will pack the electronic disk and the floppy disk:

**EDØ:,MFØ:/P** RETURN

### 3E-46. Renaming a File (/R)




3E-47. The /R FUP command option is used for renaming a file.

- Renaming has no effect on the size or location of a file. Renaming operates only on the file directory.
- An error message is displayed if the file devices specified for the new and old file names do not match.
- The no-name file is assumed when a file name is not given.

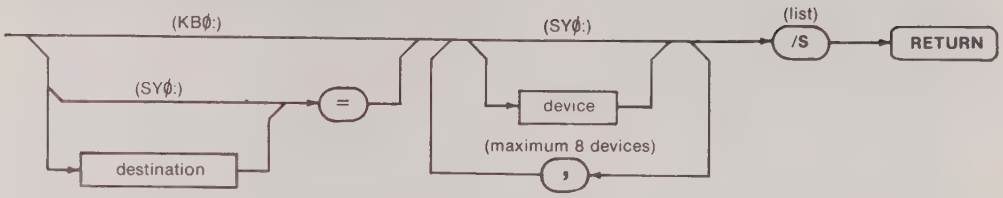
3E-48. The following example renames the file **TEST4.BAS** on the System Device to **PROG.T4**:

**PROG.T4=TEST4/R** RETURN

3E-49. The following example assigns the name **FILE84.T26** to a file on the electronic disk that previously has no name:

EDØ:FILE84.T26=EDØ:/R RETURN

### 3E-50. Scanning For Bad Blocks (/S)




3E-51. The /S FUP command option is used to scan a file-structured device for bad blocks. Each block is read and a check character is generated. It is then compared with the check character recorded with the block. Result messages are sent to the specified destination.

- Results of the scan are displayed when no destination is specified.
- Scanning a device has no effect on the integrity or location of files.

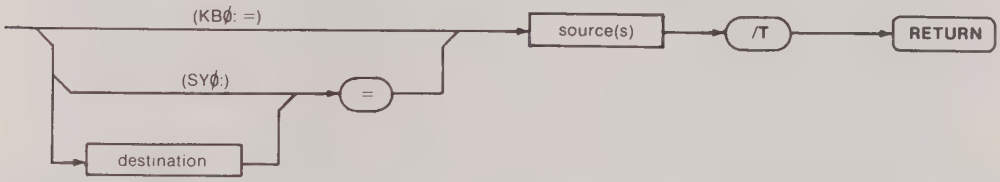
3E-52. The following example performs a scan of the System Device:

**/S** RETURN

3E-53. The following example scans the floppy disk ( **MFØ:** ) and sends the results of the scan to serial port 2 (**KB2:**)

**KB2:=MFØ:/S** RETURN

### 3E-54. Transferring ASCII files Without Error Checks (/T)




3E-55. The /T FUP command option is used to inhibit the checking for device error and for end-of-file code (⟨CTRL/Z⟩) during an ASCII file transfer.

- Except for inhibiting error checks, this command is identical to the no-option FUP command described in this section.
- If an end-of-file code is missing, it will be added to the end of the destination file.
- If a data block check-character fails, causing a device error, the error condition is ignored and the block is transferred unchanged.
- Source files are left unchanged.

3E-56. The following example will display the file **TEST6.BAS** from the System Device, including any incorrect data and regardless of a file terminator:

**TEST6/T)** RETURN

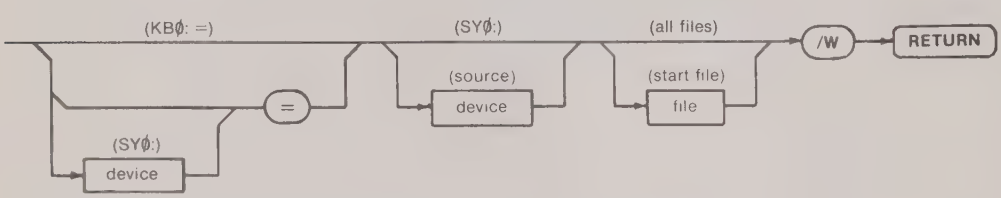
3E-57. The following example creates a back-up copy on the floppy disk ( **MFØ:** ) of the System Device file **TEST.44** which is suspected to contain an error.

**MFØ:BACKUP.T44=TEST.44/T** RETURN

3E-58. The following example copies three files from the electronic disk, which has been assigned as System Device, onto the floppy disk ( MFØ: ) for back-up purposes using the same file names.

```
MFØ:=TEST1,TEST2,TEST3/T  RETURN
```

### 3E-59. Whole Copying a File Device (/W)




3E-60. The /W FUP command option is used to transfer some or all of the files on a file device at one time using a single command. This option simplifies the task of duplicating a floppy disk, whether or not an optional electronic disk is available.

- The source and destination devices may be the floppy disk ( MFØ: ), the main memory temporary storage ( MMØ: ), or the optional electronic disk ( EDØ: ). They must however be different.

- To duplicate all or part of a floppy disk, first copy files into MMØ: or EDØ:, insert a formatted disk, and copy from MMØ: or EDØ: onto the floppy disk ( MFØ: ). More than one pass may be required, especially when using MMØ: .

- An electronic disk is needed to create a back-up floppy disk whenever any single file size exceeds the size of MMØ:, approximately 85 blocks (42K bytes). This is because the copy process works only with complete files. A two-module electronic disk is needed to create a back-up floppy disk whenever any single-file size exceeds 254 blocks (130,048 bytes).

- Files are copied in the sequence they appear on the source device directory. Copying starts at the beginning of the directory or at the specified start file.

- Binary and ASCII files may be copied within the same command.

- If the destination device already has files on it, the files copied from the source device are added. If this results in two files on the destination device with the same name, the existing file is deleted and replaced with the one copied from the source device.

- During the copy process, the names of the files being copied are displayed. If at any time the destination device does not have room for the next file to be copied, the process stops with all files complete and properly terminated. It is possible to see where the process stopped by looking at the display.

- The Whole Copy process can be properly terminated before all files are copied on the source device by typing ⟨CTRL/C⟩ during the copy of the last file desired. The file copy in progress will be completed before the operation stops.

- If ⟨CTRL/P⟩ is used to terminate a whole copy, the file copy in progress is aborted and the resulting partial file is closed.

3E-61. The following example begins a whole copy into main memory temporary storage ( **MMØ:** ) of the files on the floppy disk ( **MFØ:** ). The display should be noted for the last file copied, in the event a second pass is required.

**MMØ:=MFØ:/W** RETURN

3E-62. After noting the last file copied, insert a disk that has previously been formatted (see the /F option). The following example then copies the files onto the floppy disk ( **MFØ:** ) that were in main memory temporary storage ( **MMØ:** ).

**MFØ:=MMØ:/W** RETURN

*NOTE*

*Formatting a disk deletes the files in main memory temporary file storage.*

3E-63. If a second pass is needed, the main memory temporary storage must first be zeroed (see the /Z option) to make room for additional files. Then if a check of the directory of the original disk (see the /L option) shows the next file to be copied is **CTEST.H29**, the following example will resume whole copying with that file:

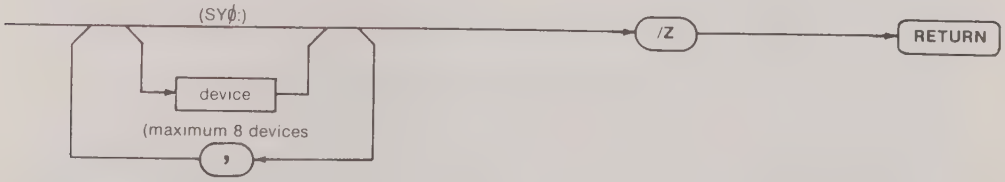
**MMØ:=MFØ:CTEST.H29/W** RETURN

3E-64. After completing a second pass of copying files into main memory ( **MMØ:** ), the new disk is reinserted. The following example then copies files onto the floppy disk ( **MFØ:** ) from main memory temporary storage ( **MMØ:** ) adding them after the files that are already in place.

**MFØ:=MMØ:/W** RETURN

3E-65. Use of the optional electronic disk ( **EDØ:** ) in place of main memory temporary storage ( **MMØ:** ) will normally speed up the process since its greater capacity will reduce the number of passes. Disk duplication using a two-module electronic disk is always a single-pass operation because the electronic disk capacity is then greater than the floppy disk.

### 3E-66. Zeroing a File Directory (/Z)




3E-67. The /Z FUP command option is used to zero the directory of one or more file devices.

- Because zeroing deletes all files from the specified device, FUP will request an affirmative before proceeding. Entries accepted as affirmative are **YES**, **Y**, **yes**, and **y**.
- The affirmative response must be on a separate line. It is required even when **/Z** is used in a command file.

3E-68. The following example zeroes the main memory temporary storage area to make room for additional files:

**MMØ:/Z** RETURN

3E-69. The following example zeros the electronic disk (**ED$\phi$:**) and main memory temporary storage (**MM$\phi$:**).

ED$\phi$:,MM$\phi$:/Z RETURN

### 3E-70. The FUP Help File

3E-71. An entry of ? given at any time in response to the FUP prompt * causes the file FUP.HLP to be displayed. As supplied on the System Disk, this file is a one-screen summary of FUP command options with examples.

- The FUP help file is a convenient quick reference of FUP command options.
- The message **?Help file not available** is displayed when the file FUP.HLP is not found on the System Device. (See Section 3A.)

### 3E-72. FUP ERROR MESSAGES

3E-73. FUP provides error messages that aid the user in recognizing faulty command inputs. Table 3E-1 summarizes and defines these errors.

**Table 3E-1. FUP Error Messages**

| MESSAGE | MEANING |
|---|---|
| ?Device Error | A non-recoverable error was detected during transfer to or from the floppy disk or the electronic disk. This also occurs when writing on an unformatted new floppy or electronic disk. |
| ?Device not ready | The floppy disk is not ready. This usually means that the disk is not inserted, or the disk drive door is not shut. |
| ?Devices do not match | A rename was attempted for files not on the same device. |
| ?File already exists | A rename was attempted using a file name already in use. |
| ?File not found | The file could not be found on the device specified. This is usually caused by misspelling, although the wrong device may have been specified. |
| ?Help file not available | A "?" was entered and the file FUP.HLP could not be located on the System Device. |
| ?Illegal option | The command option selected was not recognized. This is usually caused by typing errors. |
| ?No end-of-file | An ASCII source file was not terminated with CTRL Z. |
| ?No room for user on device | A copy or merge operation was attempted, but the resulting file would not fit on the specified device. |
| ?Not a valid device | A device was specified that is not on the list of recognized devices at the beginning of this section. this is usually caused by misspelling. |
| ?Not a valid file name | A file name in the command contains too many characters or contains characters other than letters, numbers, spaces, or "$" signs. |

**Table 3E-1. FUP Error Messages (Cont)**

| Message | MEANING |
|---|---|
| ?System error | This error should not occur under normal use. It indicates an error in FDOS or FUP. The user should make an accurate report detailing the conditions under which the error occurred and contact a Fluke Service Center. |
| ?Syntax error | The form of the command input does not match the requirements of a FUP command. This is normally caused by typing errors. |
| ?Too many files | More than eight source files were specified. |
| ?Write protected | A write operation was attempted on the floppy disk, but it has a write protect tab. Either the wrong disk is inserted, or the protection tab has to be removed. |

## 3E-74. FUP COMMAND SUMMARY

3E-75. Table 3E-2 summarizes FUP command options for quick reference.

**Table 3E-2. FUP Command Options**

| OPTION | MEANING |
|---|---|
| (omitted) | ASCII File or Data Transfer |
| /A | Assign the System Device |
| /B | Binary File Copy |
| /D | Delete File(s) |
| /E | Extended Directory List |
| /F | Format, Verify, and Zero File Device(s) |
| /L | List Directory |
| /M | Merge ASCII Files |
| /P | Pack File Device(s) |
| /R | Rename a File |
| /S | Scan File Device for Bad Blocks |
| /T | Transfer File(s) Without Error Check |
| /W | Whole Copy, Device To Device |
| /X | Exit FUP to CONMON |
| /Z | Zero File Directory |

# Section 4
# Hardware

## 4-1. INTRODUCTION

4-2. This section describes each of the hardware modules individually from a user point of view. Beginning with an overview of the hardware architecture, a separate subsection is then devoted to each module. The reader who needs additional technical detail should consult the 1720A Instrument Controller Service Manual.

## 4-3. SYSTEM ARCHITECTURE

4-4. Refer to Figure 4-1 for a functional view of the 1720A Instrument Controller. Using system software, the CPU Module processes information to and from the other modules while directing their functions. The actions of the CPU are direct responses to a variety of conditions in the sequence that they occur. Among these are the statements in a user program, inputs from the programmer keyboard or the touch sensitive display, and interrupts from the disk drive or an external device.

4-5. The modular design of the 1720A Instrument Controller results in a segregation of tasks among the modules. These modules are internally connected by a high speed parallel bus similar to that found in computer systems. In addition to ease of servicing, this design approach enhances functional and processing throughput. The subsections that follow discuss the functions of each module independent of the programming language or utility program in use. Fluke Enhanced BASIC is used wherever programming examples are provided unless specified otherwise.

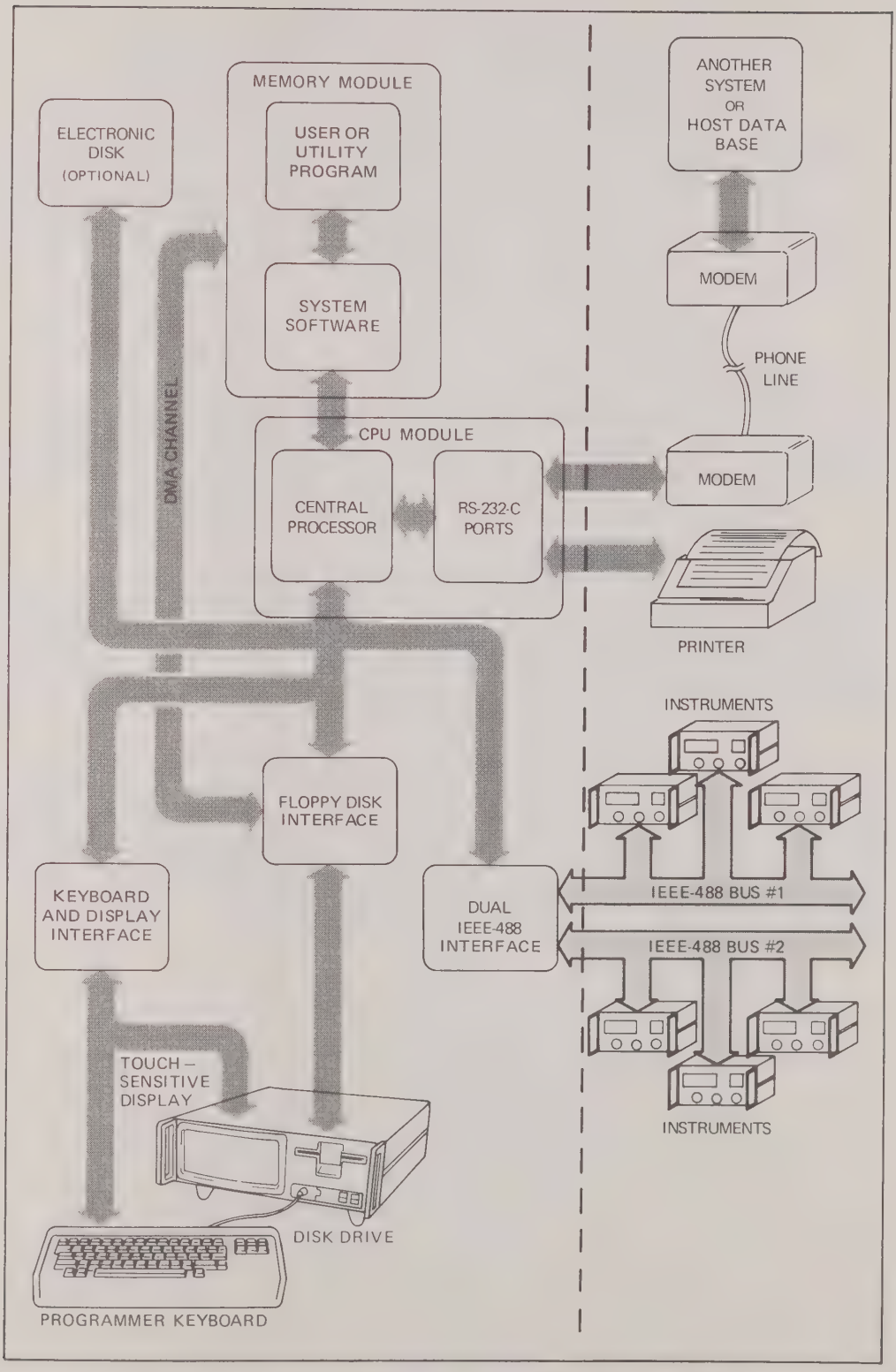



**Figure 4-1. Functional View of the 1720A Instrument Controller**

# Section 4A
# CPU MODULE

## 4A-1. INTRODUCTION

4A-2. The CPU Module gives the 1720A Instrument Controller the ability to accept and carry out user programs and control an instrument system precisely as the user intended. Through system software residing in the Main Memory Module, the user's program is processed into control actions, data collection and processing, and event-driven decisions. The system software may include processing tools such as a language interpreter as well as programming tools such as a compiler, assembler, linker, or editor. These software tools, covered in separate manuals, are placed in Main Memory by FDOS only when they are in use.

4A-3. The instructions that the CPU Module receives from memory are in 16-bit binary coded machine language corresponding to the microprocessor instruction set. Since these codes are not easily recognizable to the user,one of the functions of system software modules, themselves in binary form, is to provide the CPU with the necessary instructions to change the user recognizable utility commands and program statements into binary machine codes.

## 4A-4. LOCAL MEMORY

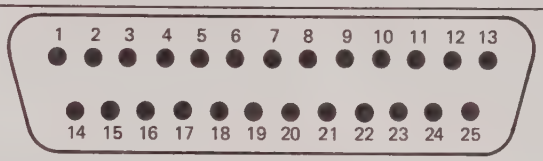
4A-5. The CPU Module intercepts memory addresses generated by its microprocessor within the upper 2048 words of address space for use on the module. Half of this is assigned to permanent read-only memory (ROM) located on the CPU module. The standard program factory recorded in this memory performs a functional test of each hardware module at start-up and then loads FDOS from the System Device into Main Memory. Since this ROM memory is in sockets, special purpose ROM components can be substituted to augment troubleshooting procedures. Refer to the 1720A Instrument Controller Service Manual for further detail.

4A-6. Of the remaining 1024 words, 256 are assigned to a local scratchpad memory, and 768 are set aside for directly addressing other modules in the system. This technique of addressing a hardware module through a memory location is called memory-mapped I/O.

## 4A-7. THE SERIAL PORTS

4A-8. The CPU Module includes two serial interfaces that conform to EIA Standard RS-232-C for Data Terminal Equipment. These ports are assigned the names KB1: and KB2: respectively by FDOS. Table 4A-1 lists the pin connections for these ports with a summary description of each pin function.

**Table 4A-1. RS-232-C Serial Data Ports (KB1: and KB2:)**




(MALE PINS)

| PIN | NAME | DIRECTION | MEANING |
|---|---|---|---|
| 1 | GND | --- | Ground. See below. |
| 2 | $\overline{\text{XMT}}$ | OUT | Transmitted Data. (Low = 1) |
| 3 | $\overline{\text{REC}}$ | IN | Received Data. (Low = 1) |
| 4 | RTS | OUT | Request To Send. Asserted (high) during data transmission. |
| 5 | CTS | IN | Clear To Send. Must be high or open for data to be transmitted. May be used to stop transmission on character boundaries by switching to low (nominal -12V). |
| 6 | DSR | IN | Data Set Ready. Not required by current driver software. |
| 7 | RET | --- | Signal Return. Zero-voltage reference point for all other signals. |
| 11 | --- | OUT | Unassigned. Always high. (+12V through 4700 ohms) |
| 12 | SCD | IN | Secondary Carrier Detect. Must be high or open for data to be transmitted. May be used to stop transmission on character boundaries by switching to low (nominal -12V). |
| 19 | SRTS | OUT | Secondary Request To Send. Always high (+12V through 4700 ohms) |
| 20 | DTR | OUT | Data Terminal Ready. Always high. (+12V through 4700 ohms) |

## 4A-9. Grounding

4A-10. Pin 1 of an RS-232-C cable supplied by Fluke is connected to the cable shield. This pin is not a reference point for signals. Its purpose is to connect the cable shield to a system ground. Make sure that use of this connection does not result in circular grounding paths (loops) through the system. Such ground loops can conduct current of enough magnitude to interfere with data transmission.

4A-11. In the 1720A, Pin 1 of Serial Port 1 is connected to Pin 1 of Serial Port 2, and then to a jumper point on the board just to the right of Serial Port 2. This jumper connects Pin 1 of both ports to the system chassis ground. Cutting the jumper disconnects both cable shields from the 1720A chassis, but they remain interconnected. In general it is good practice to ground only one end of each cable shield to avoid circular ground paths.

## 4A-12. Signal Characteristics

4A-13. RS-232-C uses larger voltages than standard logic levels, and they swing both positive and negative. Data 0 and logic true are represented by a range from +3V to +25V, nominally +12V. Data 1 and logic false are represented by a range from -3V to -25V, nominally -12V. The rate of change of these voltages is limited to 30 volts per microsecond,maximum.

4A-14. The 1720A serial port output lines are nominally ±12V. The can withstand shorting to ground or being connected to an opposing output without damage. They are designed for a DC load between 3000 ohms and 7000 ohms, with a capacitive load of less than 2500 picofarads. They are not designed for inductive loads, and cannot drive a relay coil. The capacitive loading restriction generally limits cable lengths to about fifteen meters. Special low capacitance cables, however, can extend this limit to 50 meters or more.

4A-15. The 1720A serial port input lines can withstand ±30V inputs and are set for an input threshold level of -0.75V. An open input line is interpreted as high.

## 4A-16. Parameters

4A-17. The start-up condition of the serial ports, when not modified by SET commands in a system command file, is as follows:

| | |
|---|---|
| Baud Rate | S1 setting |
| Data Bits per Character | 8 |
| Parity | None |
| Stop Bits per Character | 1 |
| End-of-Line Code | 10 (line feed) |
| End-of-File Code | 26 (⟨CTRL/Z⟩) |

4A-18. Rotary switch S1 is located on the CPU module near the connector for Serial Port 2. This switch is factory set for 300 baud. It is relatively simple to prepare a system command file containing SET commands that change any of the port parameters, independently of each other, automatically at start-up time. For this reason, the setting of S1 is normally of no concern to the user.

## 4A-19. External Terminal

4A-20. When the Video Interface Module is removed, the console device (KBϕ:) defaults to Serial Port 2. This feature allows an external terminal to be used as console if the Video Interface Module, the programmer keyboard, or the display fails. This procedure should be considered a temporary remedy only while the failure is being repaired. In this case, the setting of switch S1 becomes important.

**CAUTION**

**The internal modules contain MOS circuitry that is easily damaged by static electricity. The following procedure should be performed only by qualified service personnel familiar with the handling of MOS circuits. Specifically, the case and the technician must be grounded and interconnected. The modules must be handled only by the edges and wrapped in a conductive protective wrap when not installed. This procedure must not be performed in a low humidity environment. Failure to observe these precautions will, at the discretion of John Fluke Mfg. Co., Inc., violate the terms of the warranty as well as any service contract. Refer to the 1720A Instrument Controller Service Manual for additional information.**

4A-21. The procedure for using an external terminal as the 1720A console device (KBØ:) is:

1. Turn power OFF.

2. Remove the rear card cage cover (3 screws) using a #1 Phillips screwdriver. Facing the rear, this is the left half-panel.

3. Remove the Video Interface Module and place it in a conductive protective wrap.

4. Slide the CPU Module out approximately 3 cm.

5. Set rotary switch S1 to a baud rate that is compatible with the terminal. See Table 4A-2.

6. Press the CPU Module fully back into place.

7. Install the rear card cage cover.

8. Connect the terminal to serial port 2.

9. Turn Power ON.

10. The terminal will now be addressed by FDOS as KBØ.

**Table 4A-2. CPU S1 Settings**

| POSITION | BAUD RATE | POSITION | BAUD RATE |
|---|---|---|---|
| 0 | 110 | 4 | 1200 |
| 1 | 150 | 5 | 2400 |
| 2 | 300 | 6 | 4800 |
| 3 | 600 | 7 | 9600 |

# Section 4B
# Main Memory Module

## 4B-1. INTRODUCTION

4B-2. The Main Memory Module provides a high speed direct access storage medium for system software, user programs, and data. It is constructed of 36 dynamic RAM IC's that each store 16,384 bits, for a total capacity of 65,536 bytes or 32,768 words. Each byte includes an additional parity bit for error detection. Refreshing of the volatile dynamic storage cells is performed by on-board refresh timing circuitry that is transparent to the rest of the system. The module can transfer data in either a byte (8-bit) or word (16-bit) mode. Processor instruction fetches are in word mode, while some types of data are transferred in byte mode.

## 4B-3. MEMORY ALLOCATION

4B-4. The allocation of memory space is application dependent. The following discussion using the Fluke Enhanced BASIC Interpreter provides a useful perspective.

4B-5. Figure 4B-1 illustrates the general allocation of memory space during a running BASIC program. After studying the memory map, the trade-offs in optimizing use of this resource become apparent. Memory overflow occurs when either the data being collected by the program or an editing step attempts to use memory area in use by the interpreter stack. This stack stores return location and status information needed by the BASIC interpreter. It grows downward as required and retreats upward as tasks are completed. Variables and their data contents being used and collected by the program grow upward from the program area as needed. Should an overflow condition occur during a program run, the following may offer a solution. This discussion covers two possible causes of overflow: program size, and data overflow.

### 4B-6. Program Size

4B-7. Without allowing for variable data storage, there is room for about 24,000 characters (bytes) in a program. For an average statement length of between 10 and 15 characters, this works out to be somewhere between 1500 and 2400 program statements. If the program is that large, try restructuring it to make greater use of subroutines, or breaking it into separable tasks that can be chained together with RUN statements. Refer to the 1720A Fluke-Enhanced BASIC Programming Manual for techniques.

### 4B-8. Data Overflow

4B-9. Several options exist for the handling of data. Even if the program is going to modify the data after collection, try opening a channel to a virtual array on the floppy disk, or the optional high speed electronic disk. This will keep the data from accumulating in memory. Movements of large strings (e.g. A$=B$) will temporarily occupy the interpreter stack, an occasional cause of overflow condition during a program run. Also, see that programs do not dimension data arrays larger than the application really requires, and that space consuming string variables are de-allocated when no longer needed.

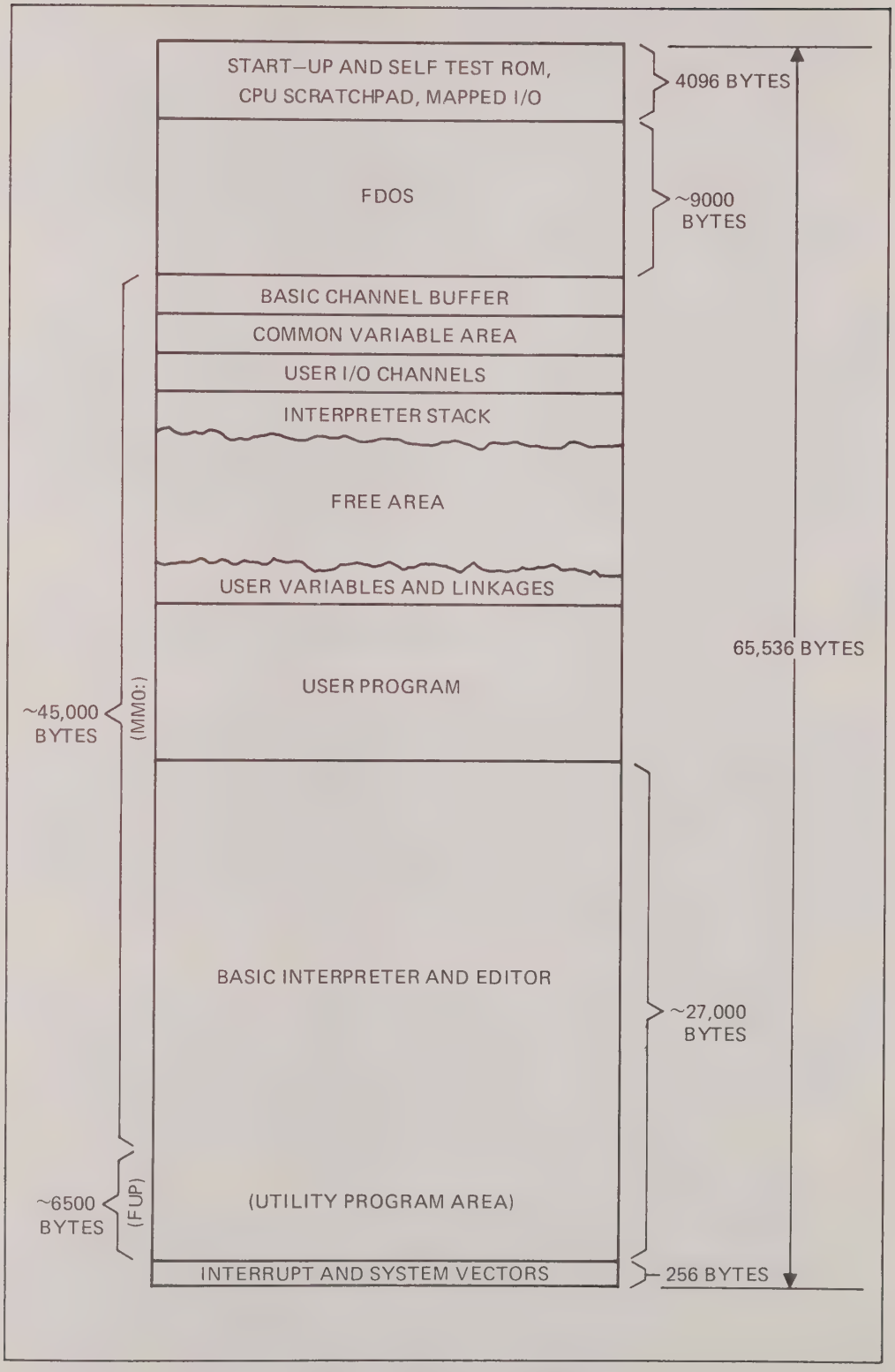



**Figure 4B-1. Memory Allocation**

# Section 4C
# Video Interface and Keyboard Module

## 4C-1. INTRODUCTION

4C-2. The Video Interface Module performs two main functions within the 1720A Instrument Controller:

1. Generates and controls all display functions in response to instructions from the CPU Module.

2. Receives inputs from the programmer keyboard and the touch sensitive panel overlaying the display, and interrupts the CPU module to pass them on.

4C-3. This module runs independent of the CPU Module, except for the passing of data and instructions. The design includes a separate microprocessor with local program and data memory, an LSI display controller, and a high speed on-board display refresh memory. The microprocessor performs a parallel scan of the programmer keyboard and the touch-sensitive display while handling screen formatting and all communications with the CPU module. The LSI display controller scans an on-board display memory, generates dot-patterns from a character ROM, and keeps the screen continually refreshed with information stored in an on-board memory. In graphics mode, character pattern information is supplied to the LSI display controller from alternate circuitry.

4C-4. On-board ROM programming for the microprocessor defines the characteristics and responses of the display to inputs from the user program. In general, these conform to subsets of recognized industrial standards. The discussion that follows covers these standards.

## 4C-5. ASCII 3.4-1968

4C-6. Table 4C-1 is a chart of ASCII codes as used by the 1720A Instrument Controller display. The left column depicts the actual display response for each ASCII code. Codes numbered decimal 0 through 31 that come from the programmer keyboard (by use of the CTRL key) are passed on to the CPU module unchanged, but are not displayed. In their place is displayed a block character (number 127) unless the code caused the CPU Module to return instructions to clear the screen. Exceptions are control codes C, P, and Z, which are displayed as ^C, ^P, and ^Z respectively.

4C-7. To send the display a code in the range of 0 through 31, it must come from the CPU Module. For example, using BASIC, the statement PRINT CHR$(27) will send the display an ESCape code. Note that the symbols available are different from the graphics mode characters presented in this section. These symbols can be enhanced with such things as high intensity or blinking just as any other character. Graphics mode characters cannot be enhanced.

## Table 4C-1. ASCII Display Chart

| 1720A DISPLAY | DECIMAL | OCTAL | HEX | 1720A DISPLAY | DECIMAL | OCTAL | HEX | 1720A DISPLAY | DECIMAL | OCTAL | HEX |
|---|---|---|---|---|---|---|---|---|---|---|---|
| NUL | 0 | 000 | 00 | , | 44 | 054 | 2C | X | 88 | 130 | 58 |
| [illegible] | 1 | 001 | 01 | - | 45 | 055 | 2D | Y | 89 | 131 | 59 |
| [illegible] | 2 | 002 | 02 | . | 46 | 056 | 2E | Z | 90 | 132 | 5A |
| [illegible] | 3 | 003 | 03 | / | 47 | 057 | 2F | [ | 91 | 133 | 5B |
| [illegible] | 4 | 004 | 04 | 0 | 48 | 060 | 30 | \ | 92 | 134 | 5C |
| [illegible] | 5 | 005 | 05 | 1 | 49 | 061 | 31 | ] | 93 | 135 | 5D |
| [illegible] | 6 | 006 | 06 | 2 | 50 | 062 | 32 | ^ | 94 | 136 | 5E |
| BEL | 7 | 007 | 07 | 3 | 51 | 063 | 33 | — | 95 | 137 | 5F |
| BS | 8 | 010 | 08 | 4 | 52 | 064 | 34 | ` | 96 | 140 | 60 |
| HT | 9 | 011 | 09 | 5 | 53 | 065 | 35 | a | 97 | 141 | 61 |
| LF | 10 | 012 | 0A | 6 | 54 | 066 | 36 | b | 98 | 142 | 62 |
| VT | 11 | 013 | 0B | 7 | 55 | 067 | 37 | c | 99 | 143 | 63 |
| FF | 12 | 014 | 0C | 8 | 56 | 070 | 38 | d | 100 | 144 | 64 |
| CR | 13 | 015 | 0D | 9 | 57 | 071 | 39 | e | 101 | 145 | 65 |
| [illegible] | 14 | 016 | 0E | : | 58 | 072 | 3A | f | 102 | 146 | 66 |
| [illegible] | 15 | 017 | 0F | ; | 59 | 073 | 3B | g | 103 | 147 | 67 |
| [illegible] | 16 | 020 | 10 | < | 60 | 074 | 3C | h | 104 | 150 | 68 |
| [illegible] | 17 | 021 | 11 | = | 61 | 075 | 3D | i | 105 | 151 | 69 |
| T | 18 | 022 | 12 | > | 62 | 076 | 3E | j | 106 | 152 | 6A |
| [illegible] | 19 | 023 | 13 | ? | 63 | 077 | 3F | k | 107 | 153 | 6B |
| [illegible] | 20 | 024 | 14 | @ | 64 | 100 | 40 | l | 108 | 154 | 6C |
| [illegible] | 21 | 025 | 15 | A | 65 | 101 | 41 | m | 109 | 155 | 6D |
| [illegible] | 22 | 026 | 16 | B | 66 | 102 | 42 | n | 110 | 156 | 6E |
| [illegible] | 23 | 027 | 17 | C | 67 | 103 | 43 | o | 111 | 157 | 6F |
| [illegible] | 24 | 030 | 18 | D | 68 | 104 | 44 | p | 112 | 160 | 70 |
| [illegible] | 25 | 031 | 19 | E | 69 | 105 | 45 | q | 113 | 161 | 71 |
| [illegible] | 26 | 032 | 1A | F | 70 | 106 | 46 | r | 114 | 162 | 72 |
| ESC | 27 | 033 | 1B | G | 71 | 107 | 47 | s | 115 | 163 | 73 |
| [illegible] | 28 | 034 | 1C | H | 72 | 110 | 48 | t | 116 | 164 | 74 |
| [illegible] | 29 | 035 | 1D | I | 73 | 111 | 49 | u | 117 | 165 | 75 |
| [illegible] | 30 | 036 | 1E | J | 74 | 112 | 4A | v | 118 | 166 | 76 |
| [illegible] | 31 | 037 | 1F | K | 75 | 113 | 4B | w | 119 | 167 | 77 |
| SPACE | 32 | 040 | 20 | L | 76 | 114 | 4C | x | 120 | 170 | 78 |
| ! | 33 | 041 | 21 | M | 77 | 115 | 4D | y | 121 | 171 | 79 |
| " | 34 | 042 | 22 | N | 78 | 116 | 4E | z | 122 | 172 | 7A |
| # | 35 | 043 | 23 | O | 79 | 117 | 4F | { | 123 | 173 | 7B |
| $ | 36 | 044 | 24 | P | 80 | 120 | 50 | \| | 124 | 174 | 7C |
| % | 37 | 045 | 25 | Q | 81 | 121 | 51 | } | 125 | 175 | 7D |
| & | 38 | 046 | 26 | R | 82 | 122 | 52 | ~ | 126 | 176 | 7E |
| ' | 39 | 047 | 27 | S | 83 | 123 | 53 | ■ | 127 | 177 | 7F |
| ( | 40 | 048 | 28 | T | 84 | 124 | 54 | | | | |
| ) | 41 | 049 | 29 | U | 85 | 125 | 55 | | | | |
| * | 42 | 050 | 2A | V | 86 | 126 | 56 | | | | |
| + | 43 | 051 | 2B | W | 87 | 127 | 57 | | | | |

4C-8. Figure 4C-1 is a short program that will display all of the displayable characters in double-size format. Type the program while in BASIC, and then type RUN followed by RETURN . The block, character number 127, is displayed in place of all control characters that are not displayable. The space character is just left of the !. Touch the screen to clear the display.

```
10   E$=CHR$(27)+"["\BL$=CHR$(127)
20   PRINT E$+"1p";CPOS(2,5);BL$:
30   FOR I=1 TO 6\PRINT CHR$(I);\NEXT I
40   FOR I=7 TO 13\PRINT BL$;\NEXT I
50   FOR I=14 TO 26\PRINT CHR$(I);\NEXT I\PRINT BL$;
60   FOR I=28 TO 31\PRINT CHR$(I);\NEXT I
70   PRINT CPOS(4,5);\FOR I=32 TO 63\PRINT CHR$(I);\NEXT I
80   PRINT CPOS(6,5);\FOR I=64 TO 95\PRINT CHR$(I);\NEXT I
90   PRINT CPOS(8,5);\FOR I=96 TO 127\PRINT CHR$(I);\NEXT I
110  END
```

**Figure 4C-1. Character Display Program**

## 4C-9. ANSI X3.64-1977

4C-10. This document of the American National Standards Institute defines some control actions to be associated with certain sequences of ASCII codes. These control sequences are always introduced by an ESCape code, usually followed by a "[" character. When sent to the Video Interface Module from the CPU Module, a control action is initiated by the Video Interface. Table 4C-2 summarizes these control actions by type, and shows the associated code sequence.

**Table 4C-2. ANSI Control Sequences**

| TYPE | ACTION | CODE SEQUENCE |
|---|---|---|
| CURSOR CONTROLS | Up n lines | ⟨ESC⟩ [ n A |
| | Down n lines | ⟨ESC⟩ [ n B |
| | Right n columns | ⟨ESC⟩ [ n C |
| | Left n columns | ⟨ESC⟩ [ n D |
| | Direct to line, column | ⟨ESC⟩ [ l ; c H |
| | Scroll down 1 line | ⟨ESC⟩ D |
| | Scroll up 1 line | ⟨ESC⟩ M |
| | Scroll to start next line | ⟨ESC⟩ E |
| ENHANCEMENTS | Off | ⟨ESC⟩ [ m |
| | High intensity | ⟨ESC⟩ [ 1 m |
| | Underline | ⟨ESC⟩ [ 4 m |
| | Blinking | ⟨ESC⟩ [ 5 m |
| | Reverse image | ⟨ESC⟩ [ 7 m |
| MODES | Normal size | ⟨ESC⟩ [ p |
| | Double size | ⟨ESC⟩ [ 1 p |
| | Graphics off | ⟨ESC⟩ [ 2 p |
| | Graphics on | ⟨ESC⟩ [ 3 p |
| | Enable keyboard | ⟨ESC⟩ [ 4 p |
| | Disable keyboard | ⟨ESC⟩ [ 5 p |

Table 4C-2. ANSI Control Sequence (cont)

| TYPE | ACTION | CODE SEQUENCE |
|---|---|---|
| ERASING | To end of line | ⟨ESC⟩ [ K |
| | To start of line | ⟨ESC⟩ [ 1 K |
| | All of line | ⟨ESC⟩ [ 2 K |
| | To end of display | ⟨ESC⟩ [ J |
| | To start of display | ⟨ESC⟩ [ 1 J |
| | All of display | ⟨ESC⟩ [ 2 J |
| CURSOR STATUS | Request cursor position | ⟨ESC⟩ [ 6 n |
| | Cursor position report | ⟨ESC⟩ [ l ; c R |
| EXTERNAL LINE CONTROL | All lines off | ⟨ESC⟩ [ q |
| | Line 1 on | ⟨ESC⟩ [ 1 q |
| | Line 2 on | ⟨ESC⟩ [ 2 q |
| | Line 3 on | ⟨ESC⟩ [ 3 q |
| | Line 4 on | ⟨ESC⟩ [ 4 q |
| | Line 5 on | ⟨ESC⟩ [ 5 q |
| | Line 6 on | ⟨ESC⟩ [ 6 q |
| | Line 7 on | ⟨ESC⟩ [ 7 q |

## 4C-11. Cursor Controls

4C-12. Cursor control sequences allow the user to move the display cursor to any position on the screen, either relative to its current position, or absolutely, by designating line and column. Scrolling commands allow movement of the entire display up or down one line at a time when movement goes beyond an edge. A cursor control sequence produces immediate results, and does not occupy a display location. Cursor controls will not move the cursor beyond the display boundaries, except that scrolling commands will cause a one-line scroll when the cursor is already at a top or bottom boundary.

4C-13. The following Fluke BASIC statement will move the cursor right 3 columns. Note the final semicolon to inhibit the usual Carriage Return and Line Feed codes following a BASIC PRINT statement.

```
PRINT CHR$(27)+"[3C";
```

## 4C-14. Enhancements

4C-15. Display enhancements are used to draw attention to a word or portion of the displayed message. Four different enhancements are available: high intensity, underlining, blinking, and reverse image. Enhancement commands occupy one character position on the display screen and define a display mode for all characters that follow until another enhancement command is encountered. Overwriting a display location containing an enhancement command will delete the enhancement. Each enhancement command cancels all previous enhancements.

4C-16. Multiple enhancements can be used simultaneously by separating the enhancement numbers with semicolons. A multiple enhancement command occupies one display location. For example:

```
PRINT CHR$(27)+'[1;4;5m";
```

4C-17. Since the refresh scanning rate of the display exceeds the 4800 baud rate at which characters are written into the display memory, underlining and reverse image commands

will momentarily cause the entire remaining display to have underlines or light background until the enhancements-off command is written. This problem can be avoided by positioning the cursor to place the enhancements-off command, then backing up to place the desired enhancement command before the start of the words to be enhanced. The use of the Display Worksheet (Part Number 533547, pad of 50) will make positioning the cursor easier. When planning an enhanced display that uses this technique, remember that the cursor spaces forward one place after each enhancement command, just as it does for any other character.

### 4C-18. Modes

4C-19. Mode selections cover three different areas: character size, the alternate graphics characters, and disabling of keyboard inputs. They may be intermixed with other display control commands, but, as explained below, the sequence can sometimes make a difference. They do not occupy a character position on the display.

4C-20. Character size is changed by controlling the display raster scan timing. For this reason it is available only as an attribute of the entire display. Character size commands clear the screen, set the entire screen to the selected character size, and move the cursor to the upper left (home) position. Normal size is 16 lines of 80 characters each. Double size is 8 lines of 40 characters each.

4C-21. If it is planned to both position the cursor and change the character display size, the program should first change the display size, and then position the cursor. This sequence is necessary because these commands return the cursor to the home position, and because cursor positioning relates to the current size mode.

4C-22. Graphics mode substitutes block graphics characters generated from alternate circuitry for the numbers 0 through 9, and the : character. Table 4C-3 shows the graphics mode characters in normal and double size. When graphics mode is enabled, these characters are displayed from alternate pattern generation circuitry whenever the numbers 0 through 9, or the : are sent to the screen. Because they are not part of the character set in the ROM, they cannot be enhanced.

4C-23. Note the names assigned to each character. Although it is not immediately obvious from looking at the character shapes, there is a correlation between single and double size characters that allow a graphics display to be set up that will change only in size and not in shape as the display mode is changed between normal and double size. This means, for example, that a display subroutine should use a 2 instead of a 6 for a normal size lower right corner. That way if the display is later changed to double size it will still be a lower right corner. Some attention to this detail will produce graphics code subroutines that are useful later for other projects.

4C-24. The program presented in Figure 4C-2 will display all of the graphics characters, first in normal size, and then, when the screen is touched, in double size. Touching the screen again will clear the display. Type it while in BASIC and then type RUN followed by RETURN.

```
10    E$=CHR$(27)+"["\CL$=E$+"2J"
20    PRINT CL$;E$+",3p";CPOS(8,24);\GOSUB 50
30    PRINT CL$;E$+"1;3p";CPOS(4,5);\GOSUB 50
40    PRINT E$+";2p";CL$\GOTO 70
50    PRINT "0 1 2 3 4 5 6 7 8 9 :"
60    WAIT FOR KEY\K$=KEY\RETURN
70    END
```

**Figure 4C-2. Graphics Character Display Program**

Table 4C-3. Graphics Mode Characters

| CHARACTER | NORMAL SIZE | DOUBLE SIZE | CHARACTER NAME |
|---|---|---|---|
| 0 | | | Top Right Corner |
| 1 | | | Top Left Corner |
| 2 | | | Bottom Right Corner |
| 3 | | | Bottom Left Corner |
| 4 | | | Top Intersect |
| 5 | | | Right Intersect |
| 6 | | | Left Intersect |
| 7 | | | Bottom Intersect |
| 8 | | | Horizontal Line |
| 9 | | | Vertical Line |
| : | | | Crossed Line |

NOTES:
1. To enable Graphics Mode, send the display ESC [ 3p
2. To disable Graphics Mode, send the display ESC [ 2p
3. In Graphics Mode, characters in the left column are displayed as shown.
4. Use the character names as defined to construct illustrations that do not change form between normal and double size.

### 4C-25. Erasing

4C-26. Commands are provided to allow a program to erase part or all of any line or of the entire display. Partial erase commands are relative to the current cursor position. An erase control sequence produces immediate results, and does not occupy a display location. For example, the BASIC statement `PRINT CHR$(27)+"[2J"` will clear the display.

### 4C-27. Cursor Status

4C-28. Two control sequences are implemented. The sequence ⟨ESC⟩[6n sent from the CPU Module to the Video Interface requests a report of the current cursor position. The Video Interface returns that report to the CPU Module in the form ⟨ESC⟩[l;cR where l = line number and c = column number.

### 4C-29. External Line Control

4C-30. Seven external lines are available for software control through control sequences from the CPU Module to the Video Interface. Two of these are located on the

Remote Interface connector on the rear of the Power Supply Module. Five are located in the connector for the programmer keyboard. The drivers of these lines are industry standard TTL, capable of sinking 24 milliamperes, or sourcing 3 milliamperes. The two lines available at the rear Remote Interface connector have an additional 390 ohm series resistor, limiting sink current to approximately 10 milliamperes when used with a red light-emitting diode that is sourced from +5 volts. Table 4C-4 shows the location of each external line by connector and pin number.

**Table 4C-4. External LED Lines**

| LINE # | CONNECTOR | PIN | LINE # | CONNECTOR | PIN |
|---|---|---|---|---|---|
| 1 | Keyboard | 7 | 5 | Keyboard | 13 |
| 2 | Keyboard | 2 | 6 | Remote Interface | 11 |
| 3 | Keyboard | 21 | 7 | Remote Interface | 13 |
| 4 | Keyboard | 8 | | | |

## 4C-31. THE TOUCH SENSITIVE DISPLAY

4C-32. The touch-sensitive panel overlaying the face of the display is constructed of two layers of transparent flexible material. The two surfaces of this material that face together are slightly metalized for conductivity, and etched into horizontal and vertical bands. The result is a resistive switch contact matrix that responds to touch pressure. Decoding circuitry in the Video/Keyboard Interface can separately identify any of 60 different areas on the display. The size of each touch sensing area or block is three double-size characters. These blocks are arranged in 6 rows of 10 each, numbered from upper left 1 through 60. The lines that the user sees on the overlay are the etch lines that form the boundaries of each touch sensitive block. A proper touch target is the center of a block.

4C-33. Operating System Software (FDOS) is notified by the Video Interface whenever a touch key has been pressed. The system variable KEY is maintained for use by other software modules, such as BASIC. This variable contains the number of the last touch key pressed, and remains set until it is read. For example, in BASIC, `K%=KEY` reads the system variable KEY.

4C-34. When a touch key is held down, it will repeat the same as a key on the programmer keyboard. The repeat rate is 15 per second. Unlike the programmer keyboard, touch sense keys have no provision for key rollover. This is the feature of a keyboard that allows a second (or more) key to be pressed before lifting the current one. Because touch keys do not have defined edges like a keyboard, multiple key rollover can cause erroneous multiple entries. Consequently, the Video Interface recognizes only one key when multiple keys are pressed.

4C-35. Figure 4C-3 illustrates the display worksheet, available in pads of 50 as Fluke part number 533547. Note that the touch sense blocks are shown overlaying both single and double-size display locations with the corresponding KEY number in the lower right corner of each block.

## 4C-36. THE PROGRAMMER KEYBOARD

4C-37. The keyboard provided with the 1720A Instrument Controller is designed primarily as a programming tool. The controller can be rack mounted easily because the keyboard is removable and there is an alternate method of operator response input through the touch-sensitive display.

4C-38. The keyboard generates the full 128-character ASCII code set. Not all of the keys are used at this time by the 1720A. This has been done to provide flexibility in future

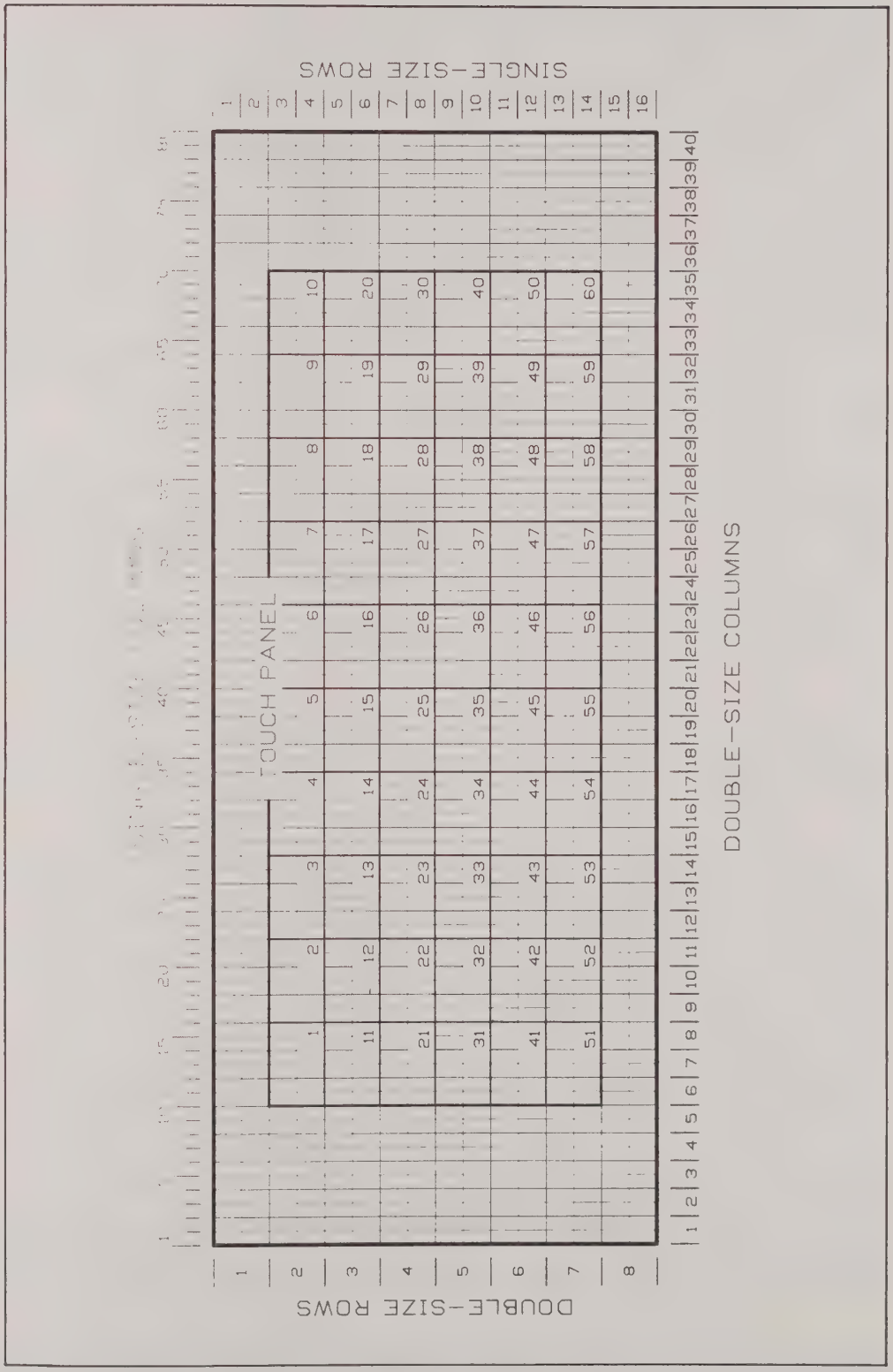



**Figure 4C-3. Display Worksheet**

applications. Two-key rollover and multiple key lockout is provided for unambiguous key response. All keys (except SHIFT and CTRL) will repeat at 15 per second when held down. An audible clicker identifies acceptance of each keystroke. The back of the keyboard case has provisions for holding its connecting cable. A cutout under the front edge provides a convenient carrying handle. Protective ridges moulded into the case allow the keyboard to be turned face down without damage to the keys.

### 4C-39. CTRL/SHIFT Keys

4C-40. The Video Interface Module interprets many keys that are modified by both CTRL and SHIFT simultaneously as additional system variable KEY inputs. These inputs are handled by FDOS in the same manner as a touch-sense key input, using the same system variable KEY. This variable contains the number of the last key input received by FDOS, and remains set until read. Table 4C-5 lists the CTRL/SHIFT key inputs and their associated KEY values. Note that touch-sense KEY values are numbered 1 through 60, and CTRL/SHIFT key inputs are numbered 61 and up. For example, in BASIC the statement `IF KEY > 60 THEN GOSUB 5000` will branch only if the last KEY entry was from a CTRL/SHIFT keyboard operation.

**Table 4C-5. CTRL/SHIFT Keys**

| CTRL/SHIFT KEY | KEY NUMBER | CTRL/SHIFT KEY | KEY NUMBER |
|---|---|---|---|
| NEXT PAGE | 61 | 6 | 91 |
| Z | 62 | 7 | 92 |
| CAPS LOCK | 64 | < | 93 |
| TAB | 65 | > | 94 |
| Q | 66 | J | 95 |
| ESC | 67 | K | 96 |
| 1 | 68 | I | 97 |
| X | 69 | O | 98 |
| C | 70 | 8 | 99 |
| A | 71 | 9 | 100 |
| S | 72 | ? | 101 |
| W | 73 | PAGE MODE | 102 |
| E | 74 | L | 103 |
| 2 | 75 | : | 104 |
| 3 | 76 | P | 105 |
| V | 77 | { | 106 |
| B | 78 | @ | 107 |
| D | 79 | SPACE | 109 |
| F | 80 | " | 111 |
| R | 81 | \ | 112 |
| T | 82 | } | 113 |
| 4 | 83 | DELETE | 114 |
| 5 | 84 | + | 115 |
| N | 85 | ~ | 116 |
| M | 86 | LINE FEED | 118 |
| G | 87 | RETURN | 119 |
| H | 88 | ← | 122 |
| Y | 89 | BACK SPACE | 123 |
| U | 90 | ↑ | 124 |

## 4C-41. COMPOSITE VIDEO OUTPUT

4C-42. The video signal that drives the 1720A display is also made available through a rear-mounted connector. This signal is intended to drive a standard data display video monitor that conforms to EIA standard RS-170. This signal complies with the timing specifications of EIA RS-170, except that equalization and serration pulses are not provided during vertical retrace. Data display monitors normally do not require this.

# Section 4D
# Floppy Disk and Interface

## 4D-1. INTRODUCTION

4D-2. The Floppy Disk Interface Module, together with the Floppy Disk Drive, provide the user with a file structured bulk storage medium that can be removed for a permanent record. Since system software is initially loaded from disk, updating a system to take advantage of newer software is simply a matter of inserting a different System Disk. The portability afforded to user programs allows the development ofstandardized test procedures that work identically for every instrument controller. The non-volatility of a disk allows the creation of permanent records for such applications as a calibration laboratory.

4D-3. In addition, the Floppy Disk Interface Module includes the Real-Time Clock used in numerous ways where time-of-day is required. The clock is discussed later in this subsection.

## 4D-4. DATA ORGANIZATION

4D-5. Programs and data on a floppy disk are organized as a sequence of 350 (512-byte) blocks. A directory is maintained in the first two blocks that enables FDOS to locate each file by its assigned name. The disk interface converts the sequential block number to a track and sector address.

4D-6. A track is one of 35 distinct radial distances from the center of the disk and is accessed by step movements of the read/write head. Data on the disk is therefore organized into concentric circles. Each track contains ten blocks or 5120 bytes of data.

4D-7. There are 10 pie-shaped radial subdivisions on a disk that divide each track into 10 sectors. An index hole punched in the disk identifies the start point for numbering the sectors. The 10 sectors on each track each contain one block or 512 bytes or data.

4D-8. Any block of data on a floppy disk can be located by positioning the head to the required track and reading sector numbers until the required one is found. Formatting a disk, one of the FUP functions presented earlier, is the process of preparing a disk for data by writing track and sector identification numbers, followed by a standard data pattern, throughout the disk surface. When reading or writing data on the disk, the DMA Floppy Interface always positions the head in or out the necessary number of steps. It then compares both track and sector identification from the disk with the desired track and sector address. When a match is found, it proceeds with the read or write operation.

4D-9. The largest file that can be put on a disk is the full capacity less two blocks for the directory. That works out to (348 blocks) X (512 bytes/block) or 178,176 bytes maximum file size. The smallest file that can be put on a disk is one block, or 512 bytes. However, the maximum number of files is limited by the size of the directory to 72.

## 4D-10. FILE ACCESS TIME

4D-11. The Floppy Disk Interface Module uses a direct memory access (DMA) technique to transfer programs and data to and from the Main Memory Module. Processor control of the internal bus is intermittently suspended in favor of the disk interface for a direct memory byte transfer. The result is a block transfer rate of 15,625 bytes per second, and an average transfer rate over a file of 25 blocks per second. This is the full transfer rate of the disk drive itself, as the DMA process easily outpaces the drive.

4D-12. If 16 seconds elapse during which no disk access has been made, the drive motor is turned off to extend its life as well as that of the head and the disk. The first access after a turn-off is delayed approximately one second while the disk comes up to speed.

4D-13. In addition, there is an average file seek time before data transfer starts, as the head moves to the proper track and locates the desired data block. Based upon an average head movement of 17.5 tracks, and an average rotation of 180 degrees, this time is about 0.55 second.

## 4D-14. FLOPPY DISKS

4D-15. Fluke makes available ten-packs of unformatted blank floppy disks which have been manufactured and certified to Fluke standards.

*NOTE*

*Performance of the 1720A Controller to published specifications can be ensured only by using Fluke Accessory Number Y1706 floppy disks, or a Fluke-approved equivalent. Contact any Fluke Sales or Service Office for information.*

4D-16. Refer to Figure 4D-1. The disk drive has a sense switch along each side of the disk slot. When fully inserted, one of them falls into the write enable notch, unless it has

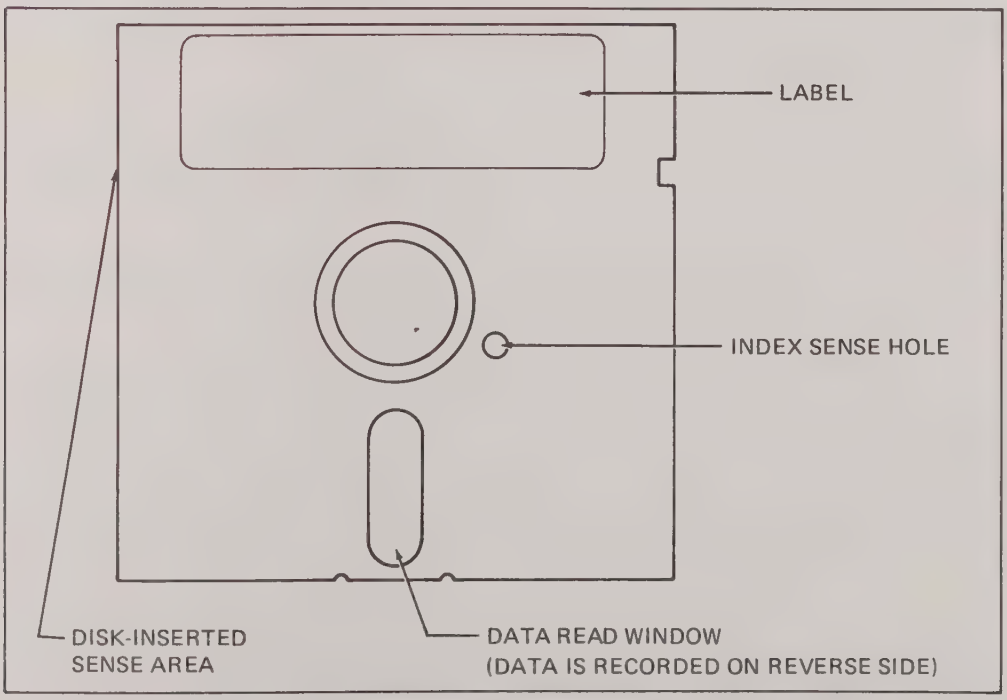



**Figure 4D-1. Floppy Disk**

been covered over with a write protect tab. The other switch along the opposite side is used to sense the presence of the disk.

4D-17. Three floppy disks are supplied with the 1720A Instrument Controller. The System Disk contains FDOS, a programming language, and the standard set of utility programs. The Diagnostic Disk contains the programs that are used by the diagnostic procedures in the 1720A Service Manual. The Data Disk is a formatted blank disk.

## 4D-18. THE REAL-TIME CLOCK

4D-19. The Floppy Disk Interface Module includes a Real-Time Clock that allows a program to label data files with date and time-of-day, to measure elapsed time, and to display or use this information in other ways. In addition, FDOS labels each directory entry with the date the file was created.

4D-20. Refer to the discussion in Section 3 on the TIME utility program for setting this function with the proper date and time. Once set, it is correct until February 29 of the next leap year, when it will have to be manually set. The internal battery will keep the clock running at least two months before requiring recharge, if the power is turned off for that long.

4D-21. Note that if an external battery is used the internal battery should be disconnected from the system. Whenever the external battery is disconnected, the clock will turn off and require resetting.

4D-22. The system can operate with the internal battery disconnected and without an external battery. In this case, it is necessary to reset the time and date, using the TIME utility, whenever the system is tuned on.

4D-23. An added feature of the Real Time Clock is a status flag that is set whenever power is first applied, and reset whenever the time and date are set. The flag is checked whenever the TIME utility is entered from within an active system command file. If it is reset, control is returned directly to CONMON, and to the next system command file line, without requiring any input. This allows a system command file to be set up with a T command that will not require a keyboard response unless power to the clock had been interrupted.

# Section 4E
# Optional Electronic Disk Modules

## 4E-1. INTRODUCTION

4E-2. The optional electronic disk is a one or two module memory system designed to provide fast access file storage. It functions just like the floppy disk, differing only in capacity and speed. Electronic disk files must be transferred to the floppy disk for permanent storage. Like a disk, it contains a file directory and must be formatted (using FUP) before it can be used. Program files must be transferred to the Memory Module before they can be executed. Data files can be directly accessed through virtual array channels.

## 4E-3. DATA ORGANIZATION

4E-4. Programs and data on the electronic disk are organized as a sequence of 256 512-byte blocks. Adding the second module, the maximum allowable, expands this to 512 blocks. This second module results in a larger electronic disk, rather than two of them. This relieves concern over which electronic disk contains a given file, and allows the storage of files larger than one electronic disk. A two-module system can handle a file larger than the capacity of the floppy disk. A single directory in the first two blocks of the first module functions as the directory for both modules.

4E-5. The largest file that can be put on the electronic disk is the full capacity less two blocks for the directory. For one module, that works out to (254 blocks) X (512 bytes/block) or 130,048 bytes. A two-module system has a maximum file capacity of (510 blocks) X (512 bytes/block) or 261,120 bytes.

4E-6. The smallest file that can be put on the electronic disk is one block, or 512 bytes. However, the number of files is limited by the size of the directory to a maximum of 72.

4E-7. Although a single module is 73% of the capacity of the floppy disk, it can be used to create back-up copies of floppy disks as long as no single file exceeds 254 blocks (130,048 bytes) in size. See the discussion of the FUP /W command in Section 3.

4E-8. An electronic disk is needed to create back-up floppy disks whenever any single-file size exceeds the memory capacity available for this function (approximately 85 blocks). A two-module electronic disk is needed to create back-up floppy disks whenever any single-file size exceeds 254 blocks (130,048 bytes).

## 4E-9. FILE ACCESS TIME

4E-10. Access time to an electronic disk file before data transfer can proceed is 18 microseconds. Thereafter data transfer proceeds at the rate of 260 blocks (133,120 bytes) per second, or at the rate the other device can accept the transfer. For a program load, which proceeds at full speed, that is equivalent to filling the entire contents of the Main Memory Module in less than 0.5 second.

## 4E-11. BATTERY BACK-UP

4E-12. The electronic disk will hold its contents intact when the 1720A Controller is powered down, provided the E-DISK BATTERY switch on the power supply is set to ENABLE and there is an internal or external source of battery power. A fully charged internal battery will support a single electronic disk module for approximately 1 hour or two electronic disk modules for about 30 minutes. Voltage sensing circuitry in the power supply determines when the battery has significantly discharged, and disconnects it from electronic disk back-up so that the internal time clock may continue to run.

4E-13. If battery back-up of the electronic disk will not be used, battery life can be extended by leaving the feature disabled. This prevents the battery from having to discharge and recharge regularly.

## 4E-14. SWITCH SETTINGS

4E-15. Electronic disk modules are identical except for a switch that identifies the module as first or second. They may be installed in either or both of the spare circuit board slots. The only requirement is that a single module must be designated as first, and that two modules must not be designated the same. Refer to Figure 4E-1, E-Disk Switch Settings.

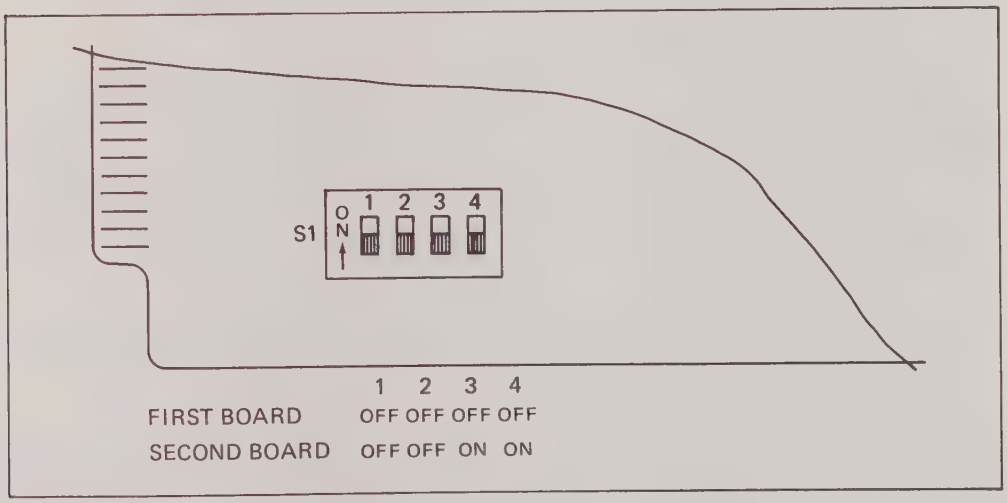



**Figure 4E-1. E-Disk Select Switch Position**

# Section 4F
# Dual IEEE-488 Interface Module

## 4F-1. INTRODUCTION

4F-2. Industrial instrumentation control is the primary design focus of the 1720A Instrument Controller. The dual IEEE-488-1978 (GPIB) interface supplied standard with each controller enables the 1720A to effectively function as two independent controllers for as many as 28 instruments on different IEEE-488 busses.

## 4F-3. FEATURES

4F-4. The availability of two separate IEEE-488 buses allows instruments to be isolated from each other. This can be a significant advantage in a number of ways. The major advantage is when the bus must connect to a device being tested. Separate buses allow the suspect device to be functionally separated from other instrumentation participating in the test. In addition, there is often a noticable speed advantage in separating devices that respond slowly to bus protocols from higher speed instruments.

4F-5. All standard controller functions defined in subsets C1 through C4 and C6 of the IEEE-488-1978 standard are implemented. Data transfer can be in any code acceptable to system software. Each port is switch set to System Controller, able to assert Interface Clear. Each port also includes readback circuitry for diagnostic self-testing.

4F-6. With current system software, this interface is designed to function as System Controller. The 1720A will not function as a listener or talker without also being System Controller. The 1720A also will not pass control to another controller on the same bus.

4F-7. Table 4F-1 lists ASCII codes and their corresponding Bus messages. Refer also to the appropriate programming language manual for Bus message sequences that have been standardized for instrument control.

**Table 4F-1. ASCII/IEEE-488-1978 Bus Codes**

| ASCII | DECIMAL | OCTAL | HEX | BINARY 7654 3210 | MESSAGE (ATN = TRUE) | | DEV. NO. |
|---|---|---|---|---|---|---|---|
| NUL | 0 | 000 | 00 | 0000 0000 | ADDRESSED COMMANDS | GTL | |
| SOH | 1 | 001 | 01 | 0000 0001 | | | |
| STX | 2 | 002 | 02 | 0000 0010 | | | |
| ETX | 3 | 003 | 03 | 0000 0011 | | | |
| EOT | 4 | 004 | 04 | 0000 0100 | | SDC PPC | |
| ENQ | 5 | 005 | 05 | 0000 0101 | | | |
| ACK | 6 | 006 | 06 | 0000 0110 | | | |
| BEL | 7 | 007 | 07 | 0000 0111 | | | |
| BS | 8 | 010 | 08 | 0000 1000 | | GET TCT | |
| HT | 9 | 011 | 09 | 0000 1001 | | | |
| LF | 10 | 012 | 0A | 0000 1010 | | | |
| VT | 11 | 013 | 0B | 0000 1011 | | | |
| FF | 12 | 014 | 0C | 0000 1100 | | | |
| CR | 13 | 015 | 0D | 0000 1101 | | | |
| SO | 14 | 016 | 0E | 0000 1110 | | | |
| SI | 15 | 017 | 0F | 0000 1111 | | | |
| DLE | 16 | 020 | 10 | 0001 0000 | UNIVERSAL COMMANDS | LLO | |
| DC1 | 17 | 021 | 11 | 0001 0001 | | | |
| DC2 | 18 | 022 | 12 | 0001 0010 | | | |
| DC3 | 19 | 023 | 13 | 0001 0011 | | | |
| DC4 | 20 | 024 | 14 | 0001 0100 | | DCL PPU | |
| NAK | 21 | 025 | 15 | 0001 0101 | | | |
| SYN | 22 | 026 | 16 | 0001 0110 | | | |
| ETB | 23 | 027 | 17 | 0001 0111 | | | |
| CAN | 24 | 030 | 18 | 0001 1000 | | SPE SPD | |
| EM | 25 | 031 | 19 | 0001 1001 | | | |
| SUB | 26 | 032 | 1A | 0001 1010 | | | |
| ESC | 27 | 033 | 1B | 0001 1011 | | | |
| FS | 28 | 034 | 1C | 0001 1100 | | | |
| GS | 29 | 035 | 1D | 0001 1101 | | | |
| RS | 30 | 036 | 1E | 0001 1110 | | | |
| US | 31 | 037 | 1F | 0001 1111 | | | |
| SP | 32 | 040 | 20 | 0010 0000 | LISTEN ADDRESSES | MLA | 0 |
| ! | 33 | 041 | 21 | 0010 0001 | | MLA | 1 |
| " | 34 | 042 | 22 | 0010 0010 | | MLA | 2 |
| # | 35 | 043 | 23 | 0010 0011 | | MLA | 3 |
| $ | 36 | 044 | 24 | 0010 0100 | | MLA | 4 |
| % | 37 | 045 | 25 | 0010 0101 | | MLA | 5 |
| & | 38 | 046 | 26 | 0010 0110 | | MLA | 6 |
| ' | 39 | 047 | 27 | 0010 0111 | | MLA | 7 |
| ( | 40 | 050 | 28 | 0010 1000 | | MLA | 8 |
| ) | 41 | 051 | 29 | 0010 1001 | | MLA | 9 |
| * | 42 | 052 | 2A | 0010 1010 | | MLA | 10 |
| + | 43 | 053 | 2B | 0010 1011 | | MLA | 11 |
| , | 44 | 054 | 2C | 0010 1100 | | MLA | 12 |
| - | 45 | 055 | 2D | 0010 1101 | | MLA | 13 |
| . | 46 | 056 | 2E | 0010 1110 | | MLA | 14 |
| / | 47 | 057 | 2F | 0010 1111 | | MLA | 15 |
| 0 | 48 | 060 | 30 | 0011 0000 | | MLA | 16 |
| 1 | 49 | 061 | 31 | 0011 0001 | | MLA | 17 |
| 2 | 50 | 062 | 32 | 0011 0010 | | MLA | 18 |
| 3 | 51 | 063 | 33 | 0011 0011 | | MLA | 19 |
| 4 | 52 | 064 | 34 | 0011 0100 | | MLA | 20 |
| 5 | 53 | 065 | 35 | 0011 0101 | | MLA | 21 |
| 6 | 54 | 066 | 36 | 0011 0110 | | MLA | 22 |
| 7 | 55 | 067 | 37 | 0011 0111 | | MLA | 23 |
| 8 | 56 | 070 | 38 | 0011 1000 | | MLA | 24 |
| 9 | 57 | 071 | 39 | 0011 1001 | | MLA | 25 |
| : | 58 | 072 | 3A | 0011 1010 | | MLA | 26 |
| ; | 59 | 073 | 3B | 0011 1011 | | MLA | 27 |
| < | 60 | 074 | 3C | 0011 1100 | | MLA | 28 |
| = | 61 | 075 | 3D | 0011 1101 | | MLA | 29 |
| > | 62 | 076 | 3E | 0011 1110 | | MLA | 30 |
| ? | 63 | 077 | 3F | 0011 1111 | | UNL | |

| ASCII | DECIMAL | OCTAL | HEX | BINARY 7654 3210 | MESSAGE (ATN = TRUE) | | DEV. NO. |
|---|---|---|---|---|---|---|---|
| @ | 64 | 100 | 40 | 0100 0000 | TALK ADDRESSES | MTA | 0 |
| A | 65 | 101 | 41 | 0100 0001 | | MTA | 1 |
| B | 66 | 102 | 42 | 0100 0010 | | MTA | 2 |
| C | 67 | 103 | 43 | 0100 0011 | | MTA | 3 |
| D | 68 | 104 | 44 | 0100 0100 | | MTA | 4 |
| E | 69 | 105 | 45 | 0100 0101 | | MTA | 5 |
| F | 70 | 106 | 46 | 0100 0110 | | MTA | 6 |
| G | 71 | 107 | 47 | 0100 0111 | | MTA | 7 |
| H | 72 | 110 | 48 | 0100 1000 | | MTA | 8 |
| I | 73 | 111 | 49 | 0100 1001 | | MTA | 9 |
| J | 74 | 112 | 4A | 0100 1010 | | MTA | 10 |
| K | 75 | 113 | 4B | 0100 1011 | | MTA | 11 |
| L | 76 | 114 | 4C | 0100 1100 | | MTA | 12 |
| M | 77 | 115 | 4D | 0100 1101 | | MTA | 13 |
| N | 78 | 116 | 4E | 0100 1110 | | MTA | 14 |
| O | 79 | 117 | 4F | 0100 1111 | | MTA | 15 |
| P | 80 | 120 | 50 | 0101 0000 | | MTA | 16 |
| Q | 81 | 121 | 51 | 0101 0001 | | MTA | 17 |
| R | 82 | 122 | 52 | 0101 0010 | | MTA | 18 |
| S | 83 | 123 | 53 | 0101 0011 | | MTA | 19 |
| T | 84 | 124 | 54 | 0101 0100 | | MTA | 20 |
| U | 85 | 125 | 55 | 0101 0101 | | MTA | 21 |
| V | 86 | 126 | 56 | 0101 0110 | | MTA | 22 |
| W | 87 | 127 | 57 | 0101 0111 | | MTA | 23 |
| X | 88 | 130 | 58 | 0101 1000 | | MTA | 24 |
| Y | 89 | 131 | 59 | 0101 1001 | | MTA | 25 |
| Z | 90 | 132 | 5A | 0101 1010 | | MTA | 26 |
| [ | 91 | 133 | 5B | 0101 1011 | | MTA | 27 |
| \ | 92 | 134 | 5C | 0101 1100 | | MTA | 28 |
| ] | 93 | 135 | 5D | 0101 1101 | | MTA | 29 |
| ^ | 94 | 136 | 5E | 0101 1110 | | MTA | 30 |
| _ | 95 | 137 | 5F | 0101 1111 | | UNT | |
| ` | 96 | 140 | 60 | 0110 0000 | SECONDARY ADDRESSES | MSA | 0 |
| a | 97 | 141 | 61 | 0110 0001 | | MSA | 1 |
| b | 98 | 142 | 62 | 0110 0010 | | MSA | 2 |
| c | 99 | 143 | 63 | 0110 0011 | | MSA | 3 |
| d | 100 | 144 | 64 | 0110 0100 | | MSA | 4 |
| e | 101 | 145 | 65 | 0110 0101 | | MSA | 5 |
| f | 102 | 146 | 66 | 0110 0110 | | MSA | 6 |
| g | 103 | 147 | 67 | 0110 0111 | | MSA | 7 |
| h | 104 | 150 | 68 | 0110 1000 | | MSA | 8 |
| i | 105 | 151 | 69 | 0110 1001 | | MSA | 9 |
| j | 106 | 152 | 6A | 0110 1010 | | MSA | 10 |
| k | 107 | 153 | 6B | 0110 1011 | | MSA | 11 |
| l | 108 | 154 | 6C | 0110 1100 | | MSA | 12 |
| m | 109 | 155 | 6D | 0110 1101 | | MSA | 13 |
| n | 110 | 156 | 6E | 0110 1110 | | MSA | 14 |
| o | 111 | 157 | 6F | 0110 1111 | | MSA | 15 |
| p | 112 | 160 | 70 | 0111 0000 | | MSA | 16 |
| q | 113 | 161 | 71 | 0111 0001 | | MSA | 17 |
| r | 114 | 162 | 72 | 0111 0010 | | MSA | 18 |
| s | 115 | 163 | 73 | 0111 0011 | | MSA | 19 |
| t | 116 | 164 | 74 | 0111 0100 | | MSA | 20 |
| u | 117 | 165 | 75 | 0111 0101 | | MSA | 21 |
| v | 118 | 166 | 76 | 0111 0110 | | MSA | 22 |
| w | 119 | 167 | 77 | 0111 0111 | | MSA | 23 |
| x | 120 | 170 | 78 | 0111 1000 | | MSA | 24 |
| y | 121 | 171 | 79 | 0111 1001 | | MSA | 25 |
| z | 122 | 172 | 7A | 0111 1010 | | MSA | 26 |
| { | 123 | 173 | 7B | 0111 1011 | | MSA | 27 |
| \| | 124 | 174 | 7C | 0111 1100 | | MSA | 28 |
| } | 125 | 175 | 7D | 0111 1101 | | MSA | 29 |
| ~ | 126 | 176 | 7E | 0111 1110 | | MSA | 30 |
| DEL | 127 | 177 | 7F | 0111 1111 | | | |

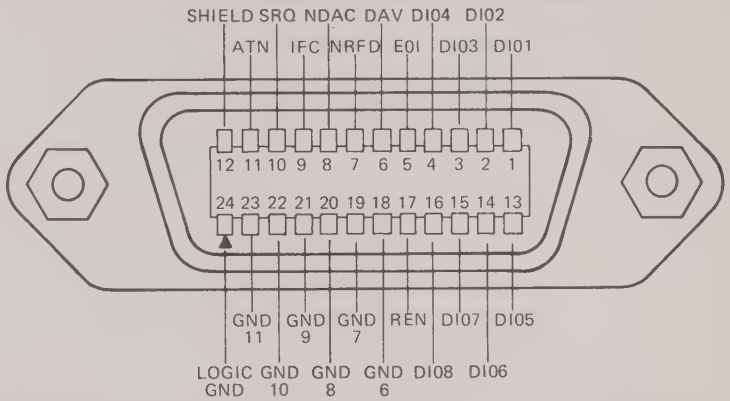
4F-8. Table 4F-2 lists the pin connections for the IEEE-488 instrument ports. These connections can be functionally divided into three groups: the Data Bus, the Management Bus, and the Transfer Bus.

4F-9. The Data Bus contains eight bidirectional active-low signal lines, DIO1 through DIO8. One byte of information (eight bits) is transferred over the bus at a time. DIO1 is least significant; DIO8 is most significant. Each byte represents a peripheral address (either primary or secondary), a control word, or a data byte.

4F-10. The Management Bus is a group of five signal lines which are used to control data transfers over the Data Bus. The Transfer Bus performs a handshake sequence executed by the talker and the listeners each time a byte is transferred over the Data Bus.

4F-11. Table 4F-3 lists the memonic abbreviations that are used in this subsection.

**Table 4F-2. IEEE-488-1978 Instrument Ports (Port 0 and Port 1)**




| BUS | PIN | SIGNAL | DEFINITION |
|---|---|---|---|
| DATA BUS | 1<br>2<br>3<br>4<br>13<br>14<br>15<br>16 | DIO1<br>DIO2<br>DIO3<br>DIO4<br>DIO5<br>DIO6<br>DIO7<br>DIO8 | Data input and output lines, bidirectional and active-low, DIO8 is most significant. Data transfers are 8-bit parallel and byte serial. |
| MANAGEMENT BUS | 11<br>23 | ATN<br>ATN Return | Attention. Activiated by the 1720A when peripheral devices are being assigned as listeners and talkers. The 1720A assumes it is the only source of this signal. |
| | 10<br>22 | SRQ<br>SRQ Return | Service request. Any peripheral device on the Instrument Bus can request the attention of the 1720A Controller by setting SRQ active low. |
| | 9<br>21 | IFC<br>IFC Return | Interface clear. Set by the 1720A to place all instruments on the bus in a predetermined reset state. The 1720A assumes that it is the only source of this signal. |
| | 17 | REN | Remote enable. Causes all responding instruments on the Bus to ignore their front panel controls and operate under remote control via signals and control messages received over the Bus. |

**Table 4F-2. IEEE-488-1978 Instrument Ports (Port 0 and Port 1) (cont)**

| BUS | PIN | SIGNAL | DEFINITION |
|---|---|---|---|
| MANAGEMENT BUS (cont) | 5 | EOI | End or identify. Can be used by a talker to identify the end of a data transfer sequence, or with ATN by a Controller, to execute a polling sequence. |
| TRANSFER BUS | 7<br>19 | NRFD<br>NRFD Return | Not ready for data. An active low signal line to indicate that one or more assinged listeners are not ready to receive the next data byte. When all of the assigned listeners for a particular data transfer have released NRFD, the NRFD line goes inactive high. The talker can then place the next data byte on the Data Bus. |
| | 6<br>18 | DAV<br>DAV Return | Data valid. Activated by the talker shortly after the placing a valid data byte on the Data Bus. An active low DAV signal tells each listener to capture the data byte presently on the Data Bus. The talker should be inhibited from activating DAV when NRFD is active low. |
| | 8<br>20 | NDAC<br>NDAC Return | Not data accepted. Held active low by each listener until the listener captures the data byte currently being transmitted over the Data Bus. When all listeners have captured the data byte, NDAC goes inactive high. This tells the talker the transfer is complete. |

**Table 4F-3. ASCII and IEEE-488 Mnemonic Abbreviations**

| | | | |
|---|---|---|---|
| ACK | Acknowledge | DLE | Data Link Escape |
| ASCII | American Standard Code for Information Interchange | ENQ | Enquiry |
| | | EOF | End of File |
| ATN | Attention | EOI | End or Indentify |
| BEL | Bell | EOT | End of Transmission |
| BS | Backspace | ESC | Escape |
| CAN | Cancel | ETB | End of Transmission Block |
| CR | Carriage Return | ETX | End of Text |
| DCL | Device Clear | FF | Form Feed |
| DCn | Device Control 1, 2, 3, or 4 | GET | Group Execute Trigger |
| DEL | Delete | GND | Ground |
| DIOn | Data Input/Output 1 through 8 | GTL | Go To Local |

**Table 4F-3. ASCII and IEEE-488 Mnemonic Abbreviations (cont)**

| | | | |
|---|---|---|---|
| HT | Horizontal Tab | PPRn | Parallel Poll Response 1 through 8 |
| IEEE | Institute of Electrical and Electronic Engineers | PPU | Parallel Poll Unconfigure |
| | | REN | Remote Enable |
| IFC | Interface Clear | RS | Record Separator |
| LF | Line Feed | SDC | Selected Device Clear |
| LLO | Local Lockout | SI | Shift In |
| MLA | My Listen Address | SO | Shift Out |
| MSA | My Secondary Address | SOH | Start of Heading |
| MTA | My Talk Address | SP | Space |
| NAK | Negative Acknowledge | SPD | Serial Poll Disable |
| NDAC | Not Data Accepted | SPE | Serial Poll Enable |
| NRFD | Not Ready For Data | SRQ | Service Request |
| NUL | Null | STB | Status Byte |
| OSA | Other Secondary Address | STX | Start of Text |
| OTA | Other Talk Address | TCT | Take Control |
| PCG | Primary Command Group | UNL | Unlisten |
| PPC | Parallel Poll Configure | UNT | Untalk |
| PPD | Parallel Poll Disable | US | Unit Separator |
| PPE | Parallel Poll Enable | | |

# Section 4G
# Remote Interface

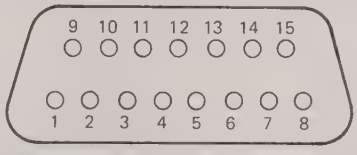
## 4G-1. INTRODUCTION

4G-2. The Remote Interface Connector provided on the rear power supply panel allows the user to do any of the following:

- Disable the front panel controls.
- Remotely perform front panel switch functions.
- Connect a larger capacity external battery.
- Halt the processor at the completion of the current machine instruction.
- Gain access to two software controlled external lines.
- Separate the battery back-up function so that it responds only to power failures and not to power turn-off by an exernal system power switch.

4G-3. Table 4G-1 lists these functions and their associated connector pin numbers.

**Table 4G-1. Remote Interface Connector (Rear Power Supply)**




| PIN | NAME | MEANING |
|---|---|---|
| 1 | ABORT SW | The front panel ABORT switch applies a ground to this pin when pressed. If the internal jumper connecting pins 1 and 2 is cut, the front panel ABORT switch is enabled by connecting this pin to pin 2. |
| 2 | ABORT ENABLE | The ABORT function is available on this pin. If the internal jumper connecting pins 1 and 2 is cut, the front panel ABORT switch is enabled by connecting this pin to pin 1. Applying ground (pin 7, 8, or 15) to this pin will cause an ABORT. |

**Table 4G-1. Remote Interface Connector (Rear Power Supply) (cont)**

| PIN | NAME | MEANING |
|---|---|---|
| 3 | RESTART SW | The front panel RESTART switch applies a ground to this pin when pressed. If the internal jumper connecting pins 3 and 4 is cut, the front panel RESTART switch is enabled by connecting this pin to pin 4. |
| 4 | RESTART ENABLE | The RESTART function is available on this pin. If the internal jumper connecting pins 3 and 4 is cut, the front panel RESTART switch is enabled by connecting this pin to pin 3. Applying ground (pin 7, 8, or 15) to this pin will cause a RESTART. |
| 5 | EXT. BATT. | Connect the positive terminal of an external battery here, if you need to extend the E-Disk back-up time provided by the internal battery. Any DC voltage between 6 and 12 may be used. Before using this function: remove the power cord, wait 5 minutes, remove the power supply, and unplug the internal battery. |
| 6 | HALT | CPU Halt. Applying ground (pin 7, 8, or 15) to this pin will halt the processor at the completion of the current machine instruction, and execute a LOAD sequence. Unless HDT software is available (see the 1720A Service Manual), an error message will result. |
| 7 | GROUND | One of three available ground pins. |
| 8 | GROUND | One of three available ground pins. |
| 9 | RESERVED | |
| 10 | N/C | |
| 11 | LED #6 | A software-controlled output line designed to drive a Light-Emitting Diode. See section 4C. |
| 12 | +5V | Five volts from the power supply. Use this to supply positive voltage for LEDs on pins 11 and 13. |
| 13 | LED #7 | A software-controlled output line designed to drive a Light-Emitting Diode. See section 4C. |
| 14 | BATTERY DISABLE | A ground applied to this line will disable the battery back-up function. Use this to prevent regular discharage of the battery when the power is applied and remove externally by a common power switch. |
| 15 | GROUND | One of three available ground lines. |

## Fluke Technical Service Centers — U.S. and Canada

### UNITED STATES

**CA, Burbank**
Fluke Technical Center
2020 N. Lincoln Street
Burbank, CA 91504
(213) 849-4641

**CA, Santa Clara**
Fluke Technical Center
2300 Walsh Avenue
Santa Clara, CA 95050
(408) 727-8121

**CO, Denver**
Fluke Technical Center
1980 S. Quebec Street, Unit 4
Denver, CO 80231
(303) 750-1228

**FL, Orlando**
Fluke Technical Center
940 N. Fern Creek Avenue
Orlando, FL 32803
(305) 896-2296

**IL, Rolling Meadows**
Fluke Technical Center
1400 Hicks Road
Rolling Meadows, IL 60008
(312) 398-5800

**MA, Waltham**
Fluke Technical Center
244 Second Avenue
Waltham, MA 02154
(617) 890-1604

**MD, Rockville**
Fluke Technical Center
5640 Fishers Lane
Rockville, MD 20852
(301) 770-1576

**MN, Apple Valley**
Fluke Technical Center
7373 West 147th Street, Suite 196
Apple Valley, MN 55124
(612) 432-7050

**NC, Greensboro**
Fluke Technical Center
1310 Beaman Place
Greensboro, NC 27408
(919) 273-1918

**NJ, Paramus**
Fluke Technical Center
West 75th Century Road
Paramus, NJ 07652
(201) 262-9550

**TX, Dallas**
Fluke Technical Center
14400 Midway Road
Dallas, TX 75240
(214) 233-9945

**WA, Mountlake Terrace**
John Fluke Mfg. Co., Inc.
21707 66th Avenue W., Suite 1
Mountlake Terrace, WA 98043
(206) 774-2206

### CANADA

**ALB, Calgary**
Allan Crawford Assoc., Ltd.
1935 30th Avenue N.E. #14
Calgary, ALB T2E 6Z5
(403) 230-1341

**BC, North Vancouver**
Allan Crawford Assoc., Ltd.
795 William Street
Burnaby, BC V5C 3H3
(604) 294-1326

**ONT, Mississauga**
Allan Crawford Assoc., Ltd.
6503 Northam Drive
Mississauga, ONT L4V 1J5
(416) 678-1500

**QUE, St. Laurent**
Allan Crawford Assoc., Ltd.
7018 Cote de Liesse
St. Laurent, QUE H4T 1E7
(514) 731-8564

## Fluke Technical Service Centers — International

**ARGENTINA**
Coasin S.A.
Virrey del Pino 4071
Buenos Aires, Argentina
Tel: 525130
TLX: 390122284 COASN AR

**AUSTRALIA**
Elmeasco Instrument Pty Ltd.
P.O. Box 30
Concord, N.S.W.
Australia 2137
Tel: (02) 736-2888
TLX: 25887

Elmeasco Instruments Pty Ltd. .
P.O. Box 107
Mt. Waverley, VIC 3149
Australia
Tel: 233-4044
TLX: 36206 (ELM VIC)

Elmeasco Instruments Pty Ltd.
Professional Suites Bldg.
G.P.O. Box 2360
Brisbane 4001
Australia
Tel: 229-3161

**AUSTRIA**
Walter Rekirsch
Elektronische Gerate GmbH & Co.
Vertriebs-KG.
Liechtensteinstrasse 97/6
A-1090 Vienna, Austria
Tel: 09-43-222-347646

**BELGIUM**
Fluke (Belgium) S.A./N.V.
6, Rue de Geneve
1140 Brussels, Belgium
Tel: 09-32-2-2164090

**BRAZIL**
Arotec S.A.
Industria e Comercio
Av. Pacaembu 811
01234 Sao Paulo S.P., Brazil
Tel: 826-2266

**CHILE**
Intronica Chile Ltda.
Casilla 16228
Santiago 9, Chile
Tel: 44940

**COLOMBIA**
Coasin Ltda.
Carrera 13, #37-43, Of. 401
Ap. Aero 29583
Bogota DE, Colombia
Tel: 285-0230

**DENMARK**
Tage Olsen A/S
Ballerup Byvej 222
DK-2750 Ballerup
Denmark
Tel: (01) 2-65 81 11

**ECUADOR**
Proteco Coasin Cia., Ltda.
Edificia "Jerico"
Ave. 12 de Octubre #2285y
Ave, Orellana (Planta Baja)
Quito, Ecuador
Tel: 526759

**EGYPT**
Lotus Engineering Organisation
P.O. Box 1252
Cairo, Egypt
Tel: 971617

**FINLAND**
Oy Findip AB
Teollisuustie 7
02700 Kauniainen
Finland
Tel: 09-358-0-502255

**FRANCE**
M.B. Electronique S.A.
Rue Fourny
ZAC de BUC
B.P. #31
78530 BUC, France
Tel: 09-33-1-9568131

**GREECE**
Hellenic Scientific
Representations Ltd.
11, Vrassida Street
Athens 615, Greece
Tel: 09-30-1-711140

## Fluke Technical Service Centers — International (cont)

**HONG KONG**
Schmidt & Co. (H.K.) Ltd.
28th Fl., Wing on Centre
111 Connaught Road
Central Hong Kong
Tel: 5-455644
TLX: 74766 SCHMC HX

**INDIA**
Hinditron Services Pvt. Ltd.
69 A.L. Jagmohandas Marg
Bombay 400 006, India
Tel: 811316
TLX: 112326 HSPL IN

Hinditron Services Pvt. Ltd.
"Hinditron House"
412 Raj Mahal Vilas Extension
Bangalore 560 006, India
Tel: 33139

Hinditron Services Pvt. Ltd.
B-36 Defence Colony
New Delhi 110024, India
Tel: 619118

**INDONESIA**
P.T. Dwi Tunggal Jaya Sakti
Sangga Buana Bldg., 1st Floor
J1, Senen Raya 44, P.O. Box 4435
Jakarta, Indonesia
Tel: 376608

**ISRAEL**
R.D.T. Electronics Engineering Ltd.
P.O. Box 75
46, Sokolov Street
Ramat Hasharon 47235
Israel
Tel: 09-972-3-483211

**ITALY**
Via Pelizza da Volpedo
22092 Ciniselio Balsamo (Milan)
Italy
Tel: 6181893
TLX: 34346

Sistrel S.P.A.
Via Guiseppe Armellini #37
00143 Rome, Italy
Tel: 09-39-6-5915551

**JAPAN**
John Fluke Mfg. Co., Inc.
Japan Branch
1 Higashikata-machi
Midori-ku
Yokohama 226, Japan
Tel: (045) 473-5425

**KOREA**
Electro-Science Korea Co.
C.P.O. Box 8446
Room 1201 Bowon Bldg.
490 Chongro - 5 Ka
Chongro-ku
Seoul, Korea
Tel: 261-7702

**MALAYSIA**
Rank O'Connor's (M) SDN, BHD,
P.O. Box 91
Petaling Jaya, Selangor
West Malaysia
Tel: 566599

**MEXICO**
C.J. Christensen S.A. De C.V.
Melchor Ocampo 150-8
Mexico 4 D.F., Mexico
Tel: (905) 535-2258

**NETHERLANDS**
Fluke (Nederland) B.V.
P.O. Box 5053
5004 EB Tillburg
The Netherlands
Tel: (013) 673973

**NEW ZEALAND**
W & K McLean Ltd.
P.O. Box 18065
Glen Innes
Auchland, New Zealand
Tel: 587-037
TLX: NZ 2763

**NORWAY**
Morgenstierne & Co. A/A
Konghellegate 3
P.O. Box 6688, Rodelokka
Oslo 5, Norway
Tel: 09-47-2-356110

**PAKISTAN**
Pak International Operations
505 Mohammadi House
1.1 Chundrigar Road
Katrachi-2 Pakistan
Tel: 221127/239052
TLX: 24494 PIOPK

**PERU**
Importaciones Y Representaciones Electronicas S.A.
Avda. Franklin D. Roosevelt 105
Lima 1, Peru
Tel: 288650

**PORTUGAL**
Equipamentos de Laboratorio, Ltda
P.O. Box 1100
Lisbon 1002, Portugal
Tel: 09-351-1-578936/547512

**REPUBLIC OF SINGAPORE**
Rank O'Connor's (PTE) Ltd.
98 Pasir Panjang Road
Singapore 0511
Republic of Singapore
Tel: 637944

**REPUBLIC OF SOUTH AFRICA**
Fluke S.A. (Pty) Ltd.
P.O. Box 39797
Bramley 2018
Republic of South Africa
Tel: (011) 786-3170

**SPAIN**
Hispano Electronica S.A.
Apartado de Correos 48
Alcorcon (Madrid)
Spain
Tel: 09-3416194108

**SWEDEN**
Teleinstrument AB
P.O. Box 490
S-162 Vallingby 4
Sweden
Tel: 09-46-8-380370

**SWITZERLAND**
Traco Electronic AG
Jenatschstrasse 1
8002 Zurich
Switzerland
Tel: 09-41-1-2010711

**TAIWAN**
CCT Associates, Inc.
P.O. Box 32-79
Taipei, Taiwan
Republic of China
Tel: (02) 391-6894/393-5760

**THAILAND**
Dynamic Supply Engineering
12, Soi Pasana 1, Ekamai
Sukhumvit 63
Bangkok 11, Thailand
Tel: 3925313

**TURKEY**
Erkman Electronik Aletler
Ticaret Anonim Sirketi
Necatiby Cad 92/3
Karakoy, Istanbul, Turkey
Tel: 09-44-1-5461

**UNITED KINGDOM**
Fluke International Corp. - UK
Colonial Way
Watford, Herts, WD24TT
United Kingdom
Tel: 09-4492340511

**URUGUAY**
Coasin Uruguaya S.R.L.
Cuareim 2196
Casilla de Correos 1400
Correo Central
Montevideo, Uruguay
Tel: 293195

**VENEZUELA**
Coasin, C.A.
Calle 9 Con Calle 4, Edif. Edinurbi
Piso 3, La Urbina
Apartado De Correos NR-70.136 Los Ruices
Caracas 1070-A Venezuela
Tel: 387842
TLX: 21027 EMVEN VE.

**WEST GERMANY**
Fluke (Deutschland) GmbH
Krausstrasse 32
8045 Ismaning
West Germany
Tel: (089) 96251

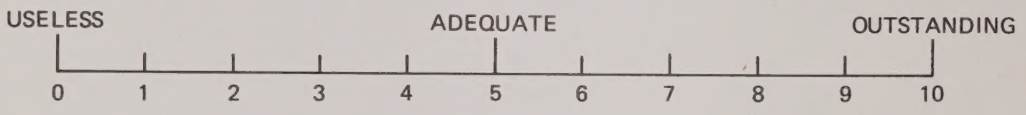
**Customers in the following countries: Bulgaria, Czechoslovakia, Hungary, Poland, Romania, U.S.S.R., and Yugoslavia, contact: Amtest Associates Ltd., P.O. Box 55, Addlestone, Surrey, KT 15 1DU, England, Tel. (0932) 52121**

**John Fluke Mfg. Co., Inc.**

**WELCOMES YOUR EVALUATION OF THIS MANUAL**

We know that the quality of this instruction manual is important to you. Help us improve that quality by giving us your inputs.

Using a scale of 0 to 10, evaluate each section of the manual or, if you prefer, only the sections that are of specific interest to you.




**Fill In Scale Rating For Each Section**

| EVALUATION AREA | SECTIONS | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | | | | | | | | |
| Accuracy | | | | | | | | | | | | |
| Sufficient Information | | | | | | | | | | | | |
| Conviently Organized | | | | | | | | | | | | |
| Clearly Written | | | | | | | | | | | | |
| Easily Understood | | | | | | | | | | | | |
| Clearly Illustrated | | | | | | | | | | | | |
| Quality of Printing and Production | | | | | | | | | | | | |

SUGGESTED IMPROVEMENTS/COMMENTS

Please print or type

Manual _____ Title _____

FROM: Name _____ Company _____

Address _____ City _____ State _____ Zip _____

Detach, Fold, Staple and Mail - No Postage Required if Mailed in United States

1720A

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 80 MOUNTLAKE TERRACE, WA

POSTAGE WILL BE PAID BY ADDRESSEE

**John Fluke Mfg. Co., Inc.**
Attention: Technical Publications
7001 220th S.W.
Mountlake Terrace, Washington 98043

www.manualsplus.com
1029152